

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-api-project-milestone-3-2024/grade/jhr4>

IT202-008-S2024 - [IT202] API Project Milestone 3 2024

Submissions:

Submission Selection

1 Submission [active] 4/27/2024 9:04:36 PM ▾

1. Instructions

2.

▲ COLLAPSE ▾

3.

4.

5.

6. Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1XE96a8DQ52Vp49XACBDTNCq0xYDt3kF29cO88E>

7. Make sure you add your ucid/date as code comments where code changes are done

8. All code changes should reach the Milestone3 branch

9. Create a pull request from Milestone3 to dev and keep it open until you get the output PDF from this assignment.

10. Gather the evidence of feature completion based on the below tasks.

Once finished, get the output PDF and copy/move it to your repository folder on your local machine.

Run the necessary git add, commit, and push steps to move it to GitHub

Complete the pull request that was opened earlier

Create and merge a pull request from dev to prod

Upload the same output PDF to Canvas

Branch name: Milestone3

Tasks: 26 Points: 10.00

● API (1 pt.)

▲ COLLAPSE ▾

● Task #1 - Points: 1

Text: Data Related to Users

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	What's the concept/association?
<input checked="" type="checkbox"/> #2	1	What sort of relationship is it (one to many, many to one, many to many, etc)
<input checked="" type="checkbox"/> #3	1	Note any other considerations

Response:

The association is favoriting articles. Users are able to favorite articles (similar to linking or just saving for later). The relationship is a many to many because the articles can be favorited by many users and the users also can favorite many articles.

Task #2 - Points: 1

Text: Updating Entities

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	When an update occurs either manually or from the API how does it affect associated data?
<input checked="" type="checkbox"/> #2	1	Do users see the old data, new data, does data need to be reassociated, etc?

Response:

When an update occurs the associated article will just show the new one. There's no change to the actual relationship as the relationship is just linked to the article id. When an update occurs to the articles, the id doesn't change so it will just show the new/updated version.

However, if the update is an enable/disable for the "is_active" of the article, it will not show the users the favorite anymore on any page they can access. There's no change to the actual association though, it just won't show up anywhere (besides on admin pages)

Handle Data Association (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)
<input checked="" type="checkbox"/> #2	1	Include uid/date comments for each code screenshot
<input checked="" type="checkbox"/> #3	1	Clearly caption screenshots

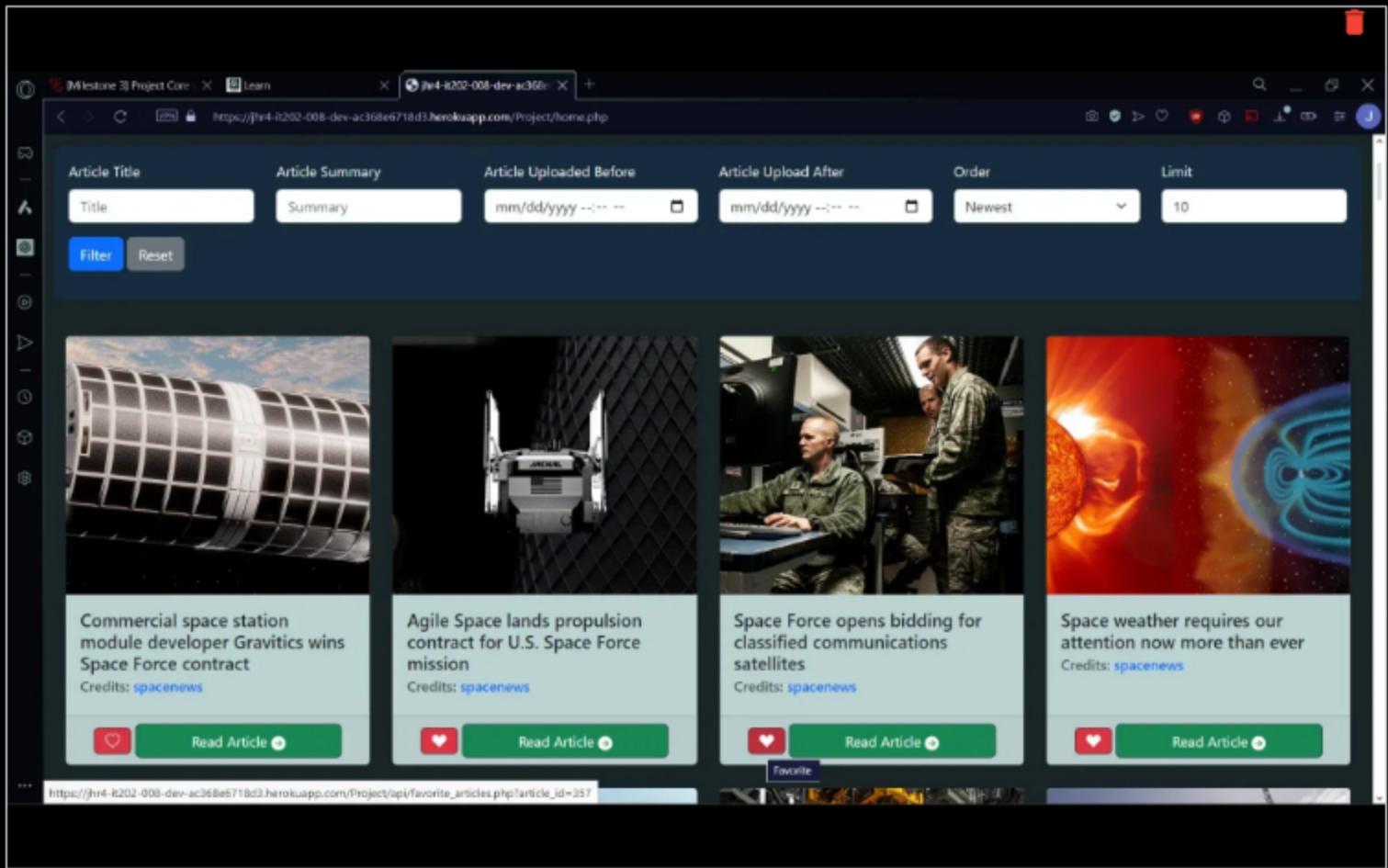
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Shows the Website from a regular user view. They can associate the article by just clicking the heart icon themselves.
Thus it will be option one.

Checklist Items (2)

#1 Option 1: Related pages will have a button to do association (like favorites or similar), Option 2: a separate page will be used to associate entities to a user by some other user (like assignment of entities)

#3 Clearly caption screenshots

A screenshot of a code editor with two files open: 'home.php' and 'favorite_articles.php'.
home.php:
A snippet of PHP code within a larger file. It includes a conditional block for adding a 'favorite' link if the current date is between April 28 and April 29.

```
178 <!--CLICK TO FAVORITE (White Heart becomes Red); IF NOT EXIST IN TABLE, LINK
179 <!--ARTICLE_ID-->
180 <a class="btn btn-danger border-dark" data-toggle="tooltip"
181 data-placement="top" title="Favorite" href="#">php echo get_url('api/
182 favorite_articles.php?article_id=' . $article['id']); ?&gt;</a
183 <?php if ((($article['user_id']) === "N/A" || $article
184 ['userArticle_isActive'] === 0) /*"N/A" b/c values being set to N/A if
185 it's null earlier*/ : ?>
186 <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
187 fill="white" class="bi-suit-heart-fill" viewBox="0 0 16 16">
188 <path d="M 1c.21 0 1.755 4 3.928 2.755 9.79 1 12 1s4 1.755 4 3.
189 92c0 3.263-3.234 4.414-7.608 9.608a.513.513 0 0 1-.784 0C3.234 9.334 0
190 8.185 0 4.92 0 2.755 1.79 3 4 1" />
191 </svg>
```

favorite_articles.php:
A snippet of PHP code within a larger file. It handles the insertion of a new record into the 'UserArticles' table if the article does not already exist.

```
1 <?php
2 //JHL 11/4/28/28
3
4 require(__DIR__ . "/../../../../lib/functions.php");
5 session_start();
6 is_logged_in(true);
7
8 //UNFAVORITE INDIVIDUAL ARTICLES
9 if(!isset($_GET['article_id']) || !is_logged_in()) { //IF NOT IN TABLE, ADD IT
10 $db = getDB();
11 $query = "INSERT INTO `UserArticles` (user_id, article_id) VALUES (:user_id, :article_id)
12 ON DUPLICATE KEY UPDATE is_active = :is_active";
13 try{
14 $stmt = $db->prepare($query);
15 $stmt->execute([':user_id'=>get_user_id(), ':article_id'=>$GET['article_id']]);
16 flash("Successfully Updated Favorite for The Article", "success");
17 }catch(PDOException $e) {
18 flash("An Error Occurred When Favoriting", "danger");
19 error_log("Error Favoriting Article: " . var_export($e, true));
20 }
21
22 //UNFAVORITE INDIVIDUAL ARTICLES
23 if(isset($_GET['article_id']) && is_logged_in())
24 $query = "DELETE FROM `UserArticles` WHERE user_id = :user_id AND article_id = :article_id";
25 try{
26 $stmt = $db->prepare($query);
27 $stmt->execute([':user_id'=>get_user_id(), ':article_id'=>$GET['article_id']]);
28 flash("Successfully Unfavorited Article", "success");
29 }catch(PDOException $e) {
30 flash("An Error Occurred When Unfavoriting", "danger");
31 error_log("Error Unfavoriting Article: " . var_export($e, true));
32 }
```

```

233 <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16">
234   <path d="M 2.748-.717-.737C5.6.281 2.514.878 1.4 3.053<-.523 1.023..>
235     6.206-6.357.955-1.806.838-3.362.314-4.385C13.486.878 10.4.28 8.717
236     2.01z#8 15C-7.333 4.868 5.279-3.84 7.824 1.145q.09.085.176.171a3 3 0 0
237     1 .176-.17C12.72-3.042 23.333 4.867 8 15" />
238 </svg>
</php endif; ?>
</a>

```

```

21 // - base if(isset($_GET['toggle_article_id'])) { //if ALREADY IN FAVORITE, REMOVE IT <- 09-21
22 $db = getDB();
23 $stmt = $db->prepare("UPDATE `UserArticles` SET is_active = :is_active WHERE user_");
24 try {
25   $stmt->execute([":user_id"=>get_user_id(), ":toggle_article_id"=>$GET["toggle_"]);
26   flash("Updated Article Favorite", "success");
27 } catch (PDOException $e) {
28   flash("An Error Occurred Toggling Favorite", "danger");
29   error_log(var_export($e->errorInfo, true));
30 }

```

Left Side is the user articles page (home page) showing the code to show the button to associate/favorite. The right side is the code for the actual change in the db upon click. UCID/Date on top of both codes.

Checklist Items (2)

#2 Include ucid/date comments for each code screenshot

#3 Clearly caption screenshots

Task #2 - Points: 1

Text: Screenshot of the association table(s)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the table(s) you made to handle the associations
<input checked="" type="checkbox"/> #2	1	Should have some example data
<input checked="" type="checkbox"/> #3	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

	Q	id int	user_id int	article_id int	is_active tinyint(1)	created timestamp	modified timestamp
> 1	1	1	30	1	2024-04-17 22:54:30	2024-04-28 18:34:50	
> 2	4	1	38	0	2024-04-17 23:26:33	2024-04-25 00:15:42	
> 3	6	7	30	1	2024-04-18 00:17:38	2024-04-28 19:15:23	
> 4	14	1	290	0	2024-04-18 02:41:05	2024-04-20 02:48:20	
> 5	16	1	271	0	2024-04-18 02:46:15	2024-04-20 02:48:20	
> 6	18	7	288	0	2024-04-18 18:47:10	2024-04-25 00:16:10	
> 7	19	1	304	0	2024-04-18 22:54:37	2024-04-28 17:01:19	

> 8	20	1	302	0	2024-04-18 22:56:00	2024-04-25 02:43:49
> 9	21	1	308	0	2024-04-18 22:56:34	2024-04-25 02:43:49
> 10	22	1	300	0	2024-04-18 23:50:54	2024-04-28 18:26:35
> 11	23	1	34	0	2024-04-19 13:50:26	2024-04-20 02:58:29
> 12	24	1	39	0	2024-04-20 02:53:40	2024-04-20 02:58:29
> 13	25	1	305	1	2024-04-21 17:33:08	2024-04-25 03:01:22

Shows the DB table of the association (Users being associated to articles). Data is shown. dev url was stated to be a mistake.

Checklist Items (4)

#1 Show the table(s) you made to handle the associations

#2 Should have some example data

#3 Make sure the heroku dev url is visible in the address bar

#4 Clearly caption screenshots

Task #3 - Points: 1

Text: Explain solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention which option your application handles regarding association
<input checked="" type="checkbox"/> #2	1	Describe each column/association table

Response:

When making an association the user and specific article are inserted into the table in a row with a 1 in the is_active table indicating active. After this to change the relationship (disable or enable again) the is_active is just set to the opposite of is_active utilizing "!is_active" (not is active). It's basically a soft delete in which the relationship is just toggled when the user wants to get unfavorite.

The user_id is the id of the user that's being associated and the article_id is the id of the article they are trying to favorite.

Created and modified just show when the association was first created and if/when it was modified.

It's also important to note the Users are able to assign themselves to a article or unassign using the heart button. Furthermore, as explained earlier the user will see the new data if the article is updated (if the update is the article being toggled off, the user won't see the article anymore on any pages).

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

<https://github.com/Jhr-4/jhr4-IT202-008/pull/63>

URL #2

<https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/Project/home>

● Current User's Association Page (2 pts.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input checked="" type="checkbox"/> #2	1	Show the single view buttons/links, delete button/links, and delete all button/link
<input checked="" type="checkbox"/> #3	1	Show variations of the number of shown items count and show the count of total number of associated items to the user
<input checked="" type="checkbox"/> #4	1	Show variations of the filter/sort including no results (proper message should be visible)
<input checked="" type="checkbox"/> #5	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Results: 5/5

Filter Reset

Unfavorite All

NASA asks the commercial space industry for a rugged, long-lived lunar rover
Credits: [astechnica](#)

Calling all space stewards: could direct-to-smartphone elevate space sustainability?
Credits: [spacenews](#)

Space weather requires our attention now more than ever
Credits: [spacenews](#)

Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars
Credits: [mars](#)

... Commercial space station module

Read Article

Read Article

Read Article

Read Article

Shows Current Users Association Page. (The articles they have favorited). Simple summary of the association/article with title and credits. Button to unfavorite (remove association) and single view article at the bottom of each card. Delete All button under filters named "Unfavorite All" Shown Item Count/Total Item count shown as "Results" in the filters card. Heroku Dev Link On top.

Checklist Items (4)

#1 Show the summary of the results with relevant information per entity

#2 Show the single view buttons/links, delete button/links, and delete all button/link

#5 Make sure the heroku dev url is visible in the address bar

#6 Clearly caption screenshots

StellarNews Home Profile My Favorites Logout

Favorited Articles

Article Title Article Summary Article Uploaded Before Article Upload After Order Favorited Limit

Samp Summary mm/dd/yyyy --:-- -- mm/dd/yyyy --:-- -- Newest 10

Results: 1/5

Unfavorite All

Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars
Credits: [mars](#)

... Commercial space station module

Read Article

Title filter and newest sort order applied showing a result of 1/5 entities.

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items to the user

#4 Show variations of the filter/sort including no results (proper message should be visible)

No Articles to Show

Favorited Articles

Article Title	Article Summary	Article Uploaded Before	Article Upload After	Order Favorited	Limit
wkuerhewourhwiehmwe	uiwehueowjwiojr	03/28/2024 10:21 PM	mm/dd/yyyy --:-- --	Oldest	10

Filter Reset Results: 0/5

Unfavorite All

Shows garbage/spam filters applied showing result of "0/5" and flash message on top of "no articles to show."

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items to the user

#4 Show variations of the filter/sort including no results (proper message should be visible)

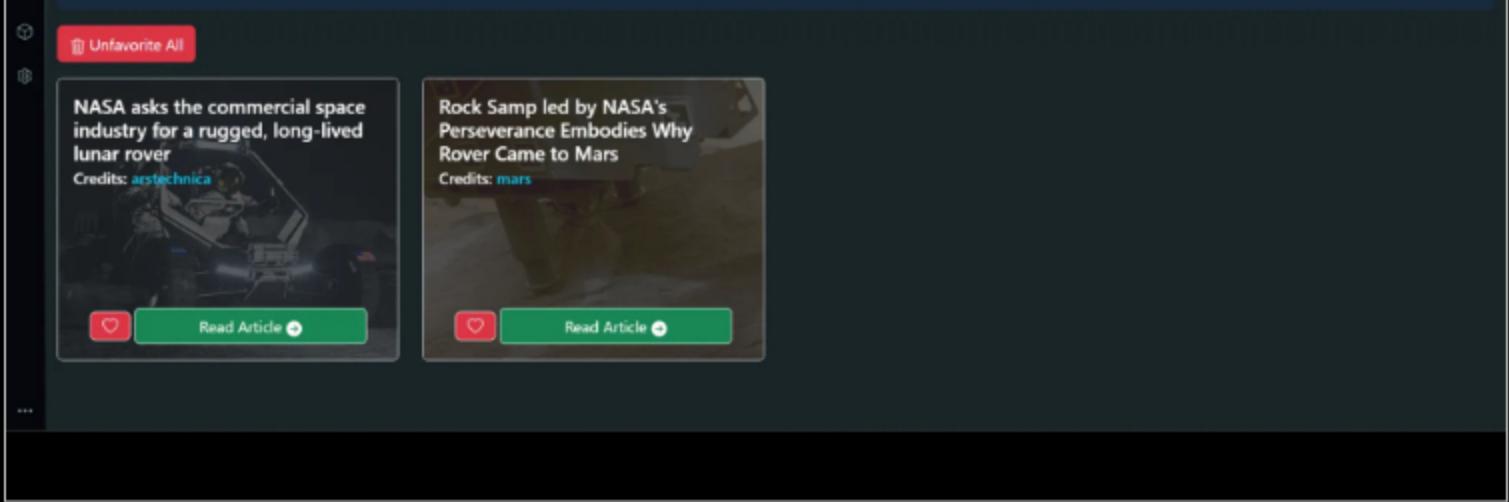
nasa

Summary

Favorited Articles

Article Title	Article Summary	Article Uploaded Before	Article Upload After	Order Favorited	Limit
nasa	Summary	mm/dd/yyyy --:-- --	mm/dd/yyyy --:-- --	Newest	2

Filter Reset Results: 2/5



Limit of 2 showing 2/5 results.

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items to the user

#4 Show variations of the filter/sort including no results (proper message should be visible)

A screenshot of the StellarNews application. At the top, there is a navigation bar with links for "Home", "Profile", "My Favorites", and "Logout". Below this is a search/filter section with fields for "Article Title", "Article Summary", "Article Uploaded Before" (with date pickers), "Article Upload After" (with date pickers), "Order Favorited" (set to "Newest"), and "Limit" (set to 10). There are "Filter" and "Reset" buttons, and a status message "Results: 0/0". A red button at the bottom left says "Unfavorite All". The main content area is a large, dark, empty space with a message "No Articles to Show" centered in it. The overall theme is dark with light-colored UI elements.

Unfavorite button in action.

Checklist Items (1)

#2 Show the single view buttons/links, delete button/links, and delete all button/link

[COLLAPSE](#)

TASK #2 - POINTS: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching the user's associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input checked="" type="checkbox"/> #5	1	Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)
<input checked="" type="checkbox"/> #6	1	Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)
<input checked="" type="checkbox"/> #7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input checked="" type="checkbox"/> #8	1	Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)
<input checked="" type="checkbox"/> #9	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #10	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

[Small](#) [Medium](#) [Large](#)

```
public_html > Project > favorites.php > ...
81 //sort and order SORT ISN'T THERE SO IT SHOULD JUST BE CREATED ALWAYS.
82 $sort = $_GET["sort", "modified", false];
83 if (!in_array($sort, ["api_id", "modified"])) {
84     $sort = "modified";
85 }
86 if ($sort === "modified" || $sort === "api_id") {
87     $sort = "UserArticles." . $sort;
88 }
89 $order = $_GET["order", "desc", false];
90 if (!in_array($order, ["asc", "desc"])) {
91     $order = "desc";
92 }
93
94 $query .= " ORDER BY $sort $order";
95
96 //LIMIT
97 try {
98     $limit = (int) $_GET["limit", "10", false];
99 } catch (PDOException $error) {
100     $limit = 10;
101 }
102 if ($limit < 1 || $limit > 100) {
103     $limit = 10;
104 }
105
106 $query .= " LIMIT $limit";
107 } else { //IF No Session data loaded OR no filters -> user will see latest articles first
108     $sort = "ArticlesTable.created";
109     $order = "desc";
110     $query .= " ORDER BY $sort $order";
111 } <- #107-111 else
112
113 $db = getDB();
114 $stmt = $db->prepare($query);
115 $results = [];
116 try {
117     $stmt->execute($params);
118     $r = $stmt->fetchAll();
119     if ($r) {
120         $results = $r;
121     }
122 } catch (PDOException $error) {
```

```
123     error_log("Error fetching articles: " . var_export($error, true));
124     flash("An error occurred", "danger"); //jhre4 || 4/28/24
125 }
```

Limit logic in the middle of screenshot. UCID / Date on bottom

Checklist Items (3)

#8 Show the logic related to filter/sort (limit should be constrained to 1-100 otherwise default to 10)

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

Shows query for getting current user related associations. Also shows more filter logic under it. UCID / Date on top.

Checklist Items (1)

#1 Show the code related to fetching the user's associations (including the query)

```
224 1--CARD ARTICLE DISPLAYING-->
225 <div class="row row-cols-1 row-cols-lg-4 row-cols-sm-2 g-4 mx-auto">
226     <?php foreach ($results as $article) : ?>
227         <?php
228             if ($article['is_active'] === "false") {
229                 continue; //Will skip the card if it's not active.
230             }
231         </?php>
232         <div class="col d-flex">
233             <div class="card bg-dark text-white border-white flex-fill w-100">
234                 " alt="Article image">
235                 <div class="card-img-overlay bg-dark opacity-75">
236                 </div>
237                 <div class="card-img-overlay d-flex flex-column">
238                     <h5 class="card-title"><?php se($article, "title", "Unknown", true) ?></h5>
239                     <!--DISPLAY LINK/SITE-->
240                     <h6 class="card-subtitle text-light">Credits:</h6>
241                     <a class="text-decoration-none text-light" href="<?php se($article, "site_url", "") ?>" target="_blank">
242                         <?php
243                             // ((https://)?(www.)?)           [-a-zA-Z0-9@%_.+~-]+@{2,256}           \.{a-z}{2,6}\b([-a-zA-Z0-9@%_.+~-]{0,256})*/?
244                             //remove https://www.   .top-domain/-----
245                         if ($article['site_url']) {
246                             if (preg_match('/((https://)?(www.)?)[-a-zA-Z0-9@%_.+~-]+@{2,256}\.{a-z}{2,6}\b([-a-zA-Z0-9@%_.+~-]{0,256})*/?', $article['site_url'])) {
247                                 $article['site_url'] = preg_replace('/((https://)?(www.)?)//', '', $article['site_url']);
248                                 $article['site_url'] = $article['site_url'].((a-z)(2,6)\b([-a-zA-Z0-9@%_.+~-]{0,256})*/?''. $article['site_url']));
249                         }
250                     </a>
251                 </div>
252             </div>
253         </div>
254     <?php endforeach; ?>
255 </div>
```

Displaying of the results as cards code. Button/link to single view on bottom Button to unfavorite (delete relationship) above the single view UCID/date on bottom.

Checklist Items (5)

#2 Show the code related to the display of the results

#3 Each record should have a button/link for single view

#4 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

Left Side is code to show button to unfavorite all. Right side shows logic of unfavoriting all and individual unfavorites/deletes. UCID/Date for top of both

Checklist Items (2)

#4 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

#5 Show the logic for deleting all associations for the user (this may be admin-only but should be present for the specific role)

Shows logic of totalShown on left. Shows logic for totalFavorited on right and the 'query' construction on the left. UCID / date on top of both.

Checklist Items (2)

#6 Show the logic related to the count of all associated items to the user (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

<input checked="" type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)
<input checked="" type="checkbox"/> #3	1	Mention the logic for single delete and delete all associations for the specific user

Response:

Logic for getting the associated data:

For this page as it shows all the associated articles it started by selecting all the columns i want to show/be able to be filtered (title, image, site_url, summary, and the image from the ArticlesTable). After this to make sure that it's an actual active article I had to add the where clause of ArticlesTable.is_active. Furthermore, as I wanted the page to only show those that are associated with the user I had to left join on the UserArticles (the association table) and only select added a where clause that allows the data to only be selected where the user id in the UsersArticle table is the same as the current users id (which is gotten from the session). It also makes sure that the relationship is active as the page should only show what's currently favorited not the ones that are unfavoirted/once favorited. The type of join also didn't really matter in this case as the where clause made sure it only selected values that I wanted (active relationship, active article, and the current users user id) so even for left join the articles that had no association were basically removed. The id of the article from the ArticlesTable is also gotten to be able to get the specific articles id allowing it to be toggled later if the user wants it (this id is displayed in the url for the button to unfavorite more discussed later).

Logic for Filters (same as MS2 basically):

For the filters its basically the same as the article list page in which the text filters use the LIKE in there where clause and the % wild card to just select everything where it includes the typed input. Furthermore, for filtering between dates by just selecting everything where the date is between what they specified. There's also a sorting option on date (newest or oldest) which was done by utilizing the created column in the ArticlesTable and ordering it by increasing dates or decreasing dates.

Deleting Specific Association:

After this came the actual displaying of the articles which was done as cards for each. For each one there's also a button to unfavorite/delete the association. This button links to another page that sends a get request by just inputting the id of the article in the url and it preforms an insert query trying to insert the data (which won't work), but it has a on duplicate clause which makes the is_active column to the opposite. If its active (1/true) it will make it (0/false) and vise versa. However, if it's on this page it means it should be already in the table and it will always be 1/active/true and upon clicking it will make it 0/false which removes it from the page.

Deleting All Associations:

There's also a delete all button which directs the user to another page which just updates the UserArticles table on rows that have their current user id (from session) and the is_active column is 1 (active/true/favorited) and toggles the is_active column by setting it to the opposite. It just negates the 1/true to false/0 and the redirects them back to the favorites page.

Shown Count:

For the shown count I just counted the number of articles in the results from the db. In other words I counted (main/outer) elements in the results array. It was originally a counter that got incremented for each article in a foreach loop, however I remembered about the count(\$array) method from which just does the same thing more simply.

Total Count:

To count the total number of results the logic was to create another query that basically the same as the current one, but with no filters that the user can apply.

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

<https://github.com/Jhr-4/jhr4-IT202-008/pull/63>

URL #2

<https://github.com/Jhr-4/jhr4-IT202-008/pull/64>

URL #3

<https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/Project/favorites.php>

All Users Association Page (likely an admin page) (2 pts.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input checked="" type="checkbox"/> #2	1	Show the single view buttons/links, delete button/links
<input checked="" type="checkbox"/> #3	1	Show the username related to the specific entity and that it's clickable
<input checked="" type="checkbox"/> #4	1	Show variations of the number of shown items count and show the count of total number of associated items
<input checked="" type="checkbox"/> #5	1	Show variations of the filter/sort including no results (proper message should be visible)
<input checked="" type="checkbox"/> #6	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #7	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web browser window with the URL <https://jhr4-it202-008-dev-ac368e6718d3.herokuapp.com/Project/admin/allFavorites.php>. The page title is "All Favorited Articles". The browser interface includes a back/forward navigation bar, search, and other standard controls.

Shows summary of results (title, credits, image) for each card. Single view (Read Article) button on bottom of each card. Usernames associated shown next to "associated users:". It's a link so it's clickable... (proof shown on the bottom the link is shown upon hover linking to profile based on id) results = shown 7 / total 7 in filter card. Dev link on top.

Checklist Items (6)

#1 Show the summary of the results with relevant information per entity

#2 Show the single view buttons/links, delete button/links

#3 Show the username related to the specific entity and that it's clickable

#4 Show variations of the number of shown items count and show the count of total number of associated items

#6 Make sure the heroku dev url is visible in the address bar

#7 Clearly caption screenshots

Shows summary of results (title, credits, image) for each card. Single view (Read Article) button on bottom of each card. Usernames associated shown next to "associated users:". It's a link so it's clickable... (proof shown on the bottom the link is shown upon hover linking to profile based on id) results = shown 7 / total 7 in filter card. Dev link on top.



Read Article



Read Article

Filter of title "mission." Results updated to shown 2/ total 7.

Checklist Items (2)

#4 Show variations of the number of shown items count and show the count of total number of associated items

#5 Show variations of the filter/sort including no results (proper message should be visible)

No Articles to Show

All Favorited Articles

Username	Article Title	Article Summary	Article Uploaded Before	Article Upload After	Order	Limit
<input type="text" value="Username"/>	<input type="text" value="eoijtowirjtoeri"/>	<input type="text" value="reojtioerjtioerjt"/>	<input type="text" value="mm/dd/yyyy"/> -- -- --	<input type="text" value="mm/dd/yyyy"/> -- -- --	<input type="button" value="Oldest"/>	<input type="text" value="1000000000"/>

Results: 0/7

No results found.

Garbage/Trash filter applied showing no results. results shown 0 / total 7

Checklist Items (2)

#4 Show variations of the number of shown items count and show the count of total number of associated items

#5 Show variations of the filter/sort including no results (proper message should be visible)

No Articles to Show

All Favorited Articles

Username	Article Title	Article Summary	Article Uploaded Before	Article Upload After	Order	Limit
<input type="text" value="Username"/>	<input type="text" value="eoijtowirjtoeri"/>	<input type="text" value="reojtioerjtioerjt"/>	<input type="text" value="mm/dd/yyyy"/> -- -- --	<input type="text" value="mm/dd/yyyy"/> -- -- --	<input type="button" value="Oldest"/>	<input type="text" value="1000000000"/>

Results: 0/7

No results found.

Username: Article Title: Article Summary: Article Uploaded Before: Article Upload After: Order: Oldest Limit: 10

Filter **Reset** Results: 3/7

Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars

Credits: mars
[3] Associated Users: [hello](#) | [hello3](#) | [hello2](#)

Heart Read Article

Star Trek vs Star Wars Debate at 39th Space Symposium: Who Won?

Credits: spacenews
[1] Associated Users: [hello](#)

Heart Read Article

Agile Space lands propulsion contract for U.S. Space Force mission

Credits: spacenews
[2] Associated Users: [hello](#) | [hello3](#)

Heart Read Article

More filters applied.. shown 3 / total 7 results.

Checklist Items (2)

#4 Show variations of the number of shown items count and show the count of total number of associated items

#5 Show variations of the filter/sort including no results (proper message should be visible)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching all associations (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Each record should have a username field that is clickable to go to the user's profile page
<input checked="" type="checkbox"/> #5	1	Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity
<input type="checkbox"/> #6	1	Show the logic related to the count of all associated items (even the ones not shown in the filtered results)
<input type="checkbox"/> #7	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input checked="" type="checkbox"/> #8	1	Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)
<input checked="" type="checkbox"/> #9	1	Include ucid/date comments for each code screenshot
<input checked="" type="checkbox"/> #10	1	Clearly section screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```

public_html > Project > admin > allFavorites.php > ...
52 //LOADING ARTICLES |hr4 || 4/28/24
53 $query = "SELECT ArticlesTable.id, title, site_url, image_url, news_text, news_summary_long, ArticlesTable.is_active, UserArticles.user_id, UserArticles.is_active AS userArticle_isActive,
54 FROM `ArticlesTable`
55 LEFT JOIN `UserArticles` ON ArticlesTable.id = UserArticles.article_id
56 LEFT JOIN Users ON Users.id = UserArticles.user_id
57 WHERE UserArticles.is_active";
58
59 $params = [];
60 $session_key = $_SERVER["SCRIPT_NAME"];
61
62 //RESET/CLEAR FILTERS BUTTON
63 $is_clear = isset($_GET["clear"]);
64 if ($is_clean) {
65     session_delete($session_key);
66     unset($_GET["clear"]);
67     redirect($session_key);
68 } else {
69     $session_data = session_load($session_key);
70 }
71
72 //FILTERING/SORTING FROM _GET
73 if (count($_GET) == 0) { //if doesn't exist
74     if ($session_data) {
75         $_GET = $session_data;
76     }
77 } <- #73-77 if (count($_GET) == 0)
78 if (count($_GET) > 0) { //if there's _GET
79     session_save($session_key, $_GET);
80     $keys = array_keys($_GET);
81
82     foreach ($form as $k => $v) {
83         if (in_array($v["name"], $keys)) {
84             $form[$k]["value"] = $_GET[$v["name"]];
85         }
86     } <- #82-86 foreach ($form as $k => $v)
87     //username
88     $username = se($_GET, "username", "", false);
89     if (!empty($username)) {
90         $query .= " AND Users.username like :username";
91         $params[":username"] = "%$username%";
92     }
93     //title
94     $title = se($_GET, "title", "", false);
95     if (!empty($title)) {

```

Code for fetching all associations query on top. Filtering logic below it. UCID / Date on top

Checklist Items (3)

#1 Show the code related to fetching all associations (including the query)

#9 Include ucid/date comments for each code screenshot

#10 Clearly caption screenshots

```

public_html > Project > admin > allFavorites.php > ...
118 //jhr4 || 4/28/24
119 //sort and order SORT ISN'T THERE SO IT SHOULD JUST BE CREATED ALWAYS.
120 $sort = se($_GET, "sort", "created", false);
121 if (!in_array($sort, ["api_id", "created"])) {
122     $sort = "created";
123 }
124 if ($sort === "created" || $sort === "api_id") {
125     $sort = "ArticlesTable." . $sort;
126 }
127 $order = se($_GET, "order", "desc", false);
128 if (!in_array($order, ["asc", "desc"])) {
129     $order = "desc";
130 }
131
132 $query .= " ORDER BY $sort $order";
133
134 //LIMIT
135 try {
136     $limit = (int)se($_GET, "limit", "10", false);
137 } catch (PDOException $error) {
138     $limit = 10;
139 }
140 if ($limit < 1 || $limit > 100) {
141     $limit = 10;
142 }

```

```

143     $query .= " LIMIT $limit";
144 } else { //If No Session data loaded && no filters -> user will see latest articles first
145     $sort = "ArticlesTable.created";
146     $order = "desc";
147     $query .= " ORDER BY $sort $order";
148 } <- #144-148 else
149
150 $db = getDB();
151 $stmt = $db->prepare($query);
152 $results = [];
153 try {
154     $stmt->execute($params);
155     $r = $stmt->fetchAll();
156     if ($r) {
157         $results = $r;
158     }
159 } catch (PDOException $error) {
160     error_log("Error fetching articles: " . var_export($error, true));
161     flash("An error occurred", "danger");
162 }

```

Logic for limiting results to 1-100 in the middle. UCID / Date on top

Checklist Items (1)

#8 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

```

86     } <- #82-86 Foreach ($form as $k -> $v)
87     //username //jhr4 || 4/28/24
88     $username = se($_GET, "username", "", false);
89     if (!empty($username)) {
90         $query .= " AND Users.username like :username";
91         $params[":username"] = "%$username%";
92     }
93     //title
94     $title = se($_GET, "title", "", false);
95     if (!empty($title)) {
96         $query .= " AND title like :title";
97         $params[":title"] = "%$title%";
98     }
99     //summary
100    $summary = se($_GET, "summary", "", false);
101    if (!empty($summary)) {
102        $query .= " AND news_summary_long like :summary";
103        $params[":summary"] = "%$summary%";
104    }
105
106    //date range
107    $MAX_timestamp = se($_GET, "MAX_timestamp", "", false);
108    if (!empty($MAX_timestamp) && $MAX_timestamp >= 0) {
109        $query .= " AND ArticlesTable.created <= :MAX_timestamp";
110        $params[":MAX_timestamp"] = $MAX_timestamp;
111    }
112    $MIN_timestamp = se($_GET, "MIN_timestamp", "", false);
113    if (!empty($MIN_timestamp) && $MIN_timestamp >= 0) {
114        $query .= " AND ArticlesTable.created <= :MIN_timestamp";
115        $params[":MIN_timestamp"] = $MIN_timestamp;
116    }
117
118    //sort and order SORT ISN'T THERE SO IT SHOULD JUST BE CREATED
119    $sort = se($_GET, "sort", "created", false);
120    if (!in_array($sort, ["api_id", "created"])) {
121        $sort = "created";
122    }
123    if ($sort == "created" || $sort == "api_id") {
124        $sort = "ArticlesTable." . $sort;
125    }
126    $order = se($_GET, "order", "desc", false);
127    if (!in_array($order, ["asc", "desc"])) {
128        $order = "desc";
129    }

```

Logic for Username Sort. UCID / Date on top.

Checklist Items (1)

#8 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

```

html> Project> admin > # all_favorites.php > @div>row>row>col-1>row>col-lg-4>row>col-sm-2>g-4 mx-auto > @form > @div>card flex > @div>card bg-dark text-white border-white flex-sm-limit-3 > @div>card-img-overlay d-flex flex-column > @p>text-light > @input>user_1
<div class="row row-cols-1 row-cols-lg-4 row-cols-sm-2 g-4 mx-auto">
<form method="POST">
<div class="col d-flex">
<div class="card bg-dark text-white border-white flex-fill mt-3">
<div class="card-img-overlay d-flex flex-column"> <!--jhr4 || 4/28/24-->
<!--DISPLAY TITLE-->
<h5 class="card-title">
<?php if ($article['is_active'] === "false") : ?>
<span class="text-danger">(Disabled)</span><?php se($article, "title", "Unknown", true); ?>
<?php else : ?>
<?php se($article, "title", "Unknown", true); ?>
<?php endif; ?>
</h5>
<!--DISPLAY LINK/SITE-->
<a href=""><?php se($article, "site_url", "")?><target=_blank>

```

Code related to displaying the results. Button to single view on bottom. Unfavorite button above it. Username of each user is clickable above the button "Associated Users:" links to profile (Page it self is an admin page so no logic to only allow only admins to control delete) UCID/ Date on top.

Checklist Items (4)

#2 Show the code related to the display of the results

#3 Each record should have a button/link for single view

#4 Each record should have a username field that is clickable to go to the user's profile page

#5 Each record should have a button/link for delete (this may be an admin-only thing but should be present for the specific role) Note: this is to delete the relationship and not the specific entity

Total Favorited (Left Side Top + Right side) and Shown Favorited (Left Side Top) code. UCID / Date on top of both

Checklist Items (2)

#6 Show the logic related to the count of all associated items (even the ones not shown in the filtered results)

#7 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention how you determine the result list (include the association logic and filters)
<input checked="" type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)
<input checked="" type="checkbox"/> #3	1	Mention the logic for single delete
<input checked="" type="checkbox"/> #4	1	Mention the logic for handling the username requirements

Response:

For this page as it shows all the associated users I selected the data/columns (same as the page before) I want from the ArticleTable and then left joined with the userarticle table where the id of the article is the same as the associated articles, it also left joins onto the users table where the user id matches the user_id column in the association table. I also make sure that the association in the userarticle table is active. Basically originally it was the same as the other page without the Users table and the specific user id. However, I needed to get the usernames associated so I left joined the users based on the user_id's in the association table to allow the usernames to be gotten. Thus, I also add to the original select the username from the Users table and the id of the user.

This also query also resulted in data of duplicates as many users can be associated to the same article. However, I couldn't just do DISTINCT articles because I needed the information for each user (username and id to create a link to their profile).

Therefore, I created I made a loop going though each of the data/row/article from the query and created another list that stored all the article id's (which are distinct or should be) and stores the associated username and their id as an associative array. Furthermore, if the article id already exists in this other list, it means that there was a duplicate article (another user also favorited it) so it takes the id and username associated with it and just unsets that article. Therefore, after the loop the articles that are associated with at least 1 user are in the results and there's an second array with the article titles and the users associated with it which will be used to display the link to the profiles and the associated user names.

For the shown item count I implemented it after the unsetting of duplicates to allow for it to be accurate. It just counts the number of articles/entities in the results array with count() method.

For the total item count as there was a problem with the duplicate articles, I utilized the same query again (no filters) and added a DISTINCT after the select to make sure that only distinct articles are selected.

On this page I created a delete for each article as its own POST form. For each user associated there's a checkbox that has the id of the user as the value. There's also an hidden field that has the value of the article id. On clicking the unfavorate icon the data of the selected users is sent and also the article id from the hidden input value is sent as a POST request/data which is processed in a pretty bad way by just looping though each user and toggling the is_active to the opposite. In this case as all the is_actives are already set to 1 it just sets it to 0/false removing it from the page after when the fetching is done.

For filtering on username and the title it's basically the same in which it selects everything that has what the user inputs somewhere with the % wildcard. It's also a feature when you type a specific user's name it just shows their articles and only their name for the articles (not the other users who are associated).

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

<https://github.com/Jhr-4/jhr4-IT202-008/pull/65>

URL #2

<https://github.com/Jhr-4/jhr4-IT202-008/pull/66>

URL #3

<https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/Project/admin/allFavorites.php>

Unassociated Page (2 pts.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of this page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the summary of the results with relevant information per entity
<input checked="" type="checkbox"/> #2	1	Show the single view buttons/links
<input checked="" type="checkbox"/> #3	1	Show variations of the number of shown items count and show the count of total number of associated items
<input checked="" type="checkbox"/> #4	1	Show variations of the filter/sort including no results (proper message should be visible)
<input checked="" type="checkbox"/> #5	1	Make sure the heroku dev url is visible in the address bar

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Shows Summary of Articles (Title, Credits, Image). Button to view on bottom. Dev Link on Top.

Checklist Items (4)

#1 Show the summary of the results with relevant information per entity

#2 Show the single view buttons/links

#5 Make sure the heroku dev url is visible in the address bar

#6 Clearly caption screenshots

qwaeijqwaiejqwoej rjgrheatherothjer mm/dd/yyyy --:-- -- mm/dd/yyyy --:-- -- Date Newest 10

Filter Reset Results: 0/93

No Articles Found.

Garbage/Spam Filters No Articles Found. 0/93 Results

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items

#4 Show variations of the filter/sort including no results (proper message should be visible)

Milestone 3 Project Core | Learn https://jh4-r252-008-dev-a368e6718d3.herokuapp.com/Project/admin/not_favorited.php?title=mars&summary=&MAX_timestamp=&MIN_timestamp=&sort=create

StellarNews Home Profile My Favorites Manage Roles Manage Articles Manage Favorites Logout

Not Favorited Articles

Article Title	Article Summary	Article Uploaded Before	Article Upload After	Sort	Order	Limit
<input type="text" value="mars"/>	<input type="text" value="Summary"/>	<input type="text" value="mm/dd/yyyy --:-- --"/>	<input type="text" value="mm/dd/yyyy --:-- --"/>	<input type="button" value="Date"/>	<input type="button" value="Oldest"/>	<input type="text" value="3"/>

Results: 3/93

ESA awards contract to Thales Alenia Space to restart ExoMars
Credits: [spacenews](#)

ESA Awards €522M ExoMars 2028 Contract to Thales Alenia Space
Credits: [europeanspaceflight](#)

NASA Invites Media to Mars Sample Return Update
Credits: [mars](#)

Shows Title/Limit Filter. Results 3 shown / 93 total.

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items

#4 Show variations of the filter/sort including no results (proper message should be visible)

Screenshot of a web application titled "StellarNews" showing "Not Favorited Articles". The interface includes a search bar, navigation links (Home, Profile, My Favorites, Manage Roles, Manage Articles, Manage Favorites, Logout), and a filter section with fields for Article Title, Article Summary, Article Uploaded Before (04/20/2024 02:23 PM), Article Upload After (mm/dd/yyyy), Sort (Date, Oldest), and Limit (10). The results count is 7/93. Below the filters are seven article cards:

- Investing in Space: NASA's tool for seeing the solar eclipse** (Credits: cnbc) - Read Article
- NASA asks the commercial space industry for a rugged, long-lived lunar rover** (Credits: arstechnica) - Read Article
- NASA Picks Three Companies for Lunar Terrain Vehicle Feasibility Studies** (Credits: spacepolicyonline) - Read Article
- NASA selects three companies to advance Artemis lunar rover designs** (Credits: spacenews) - Read Article
- NASA to Develop a Framework for Space Sustainability** (Credits: spacepolicyonline) - Read Article
- NASA rolls out new space sustainability strategy** (Credits: spacenews) - Read Article
- NASA Invites Media to Mars Sample Return Update** (Credits: mars) - Read Article

Name Filter + Date filter + Limit Filter Results: 7/93

Checklist Items (2)

#3 Show variations of the number of shown items count and show the count of total number of associated items

#4 Show variations of the filter/sort including no results (proper message should be visible)

Task #2 - Points: 1

Text: Screenshot the code

Checklist

*The checkboxes are for your own tracking

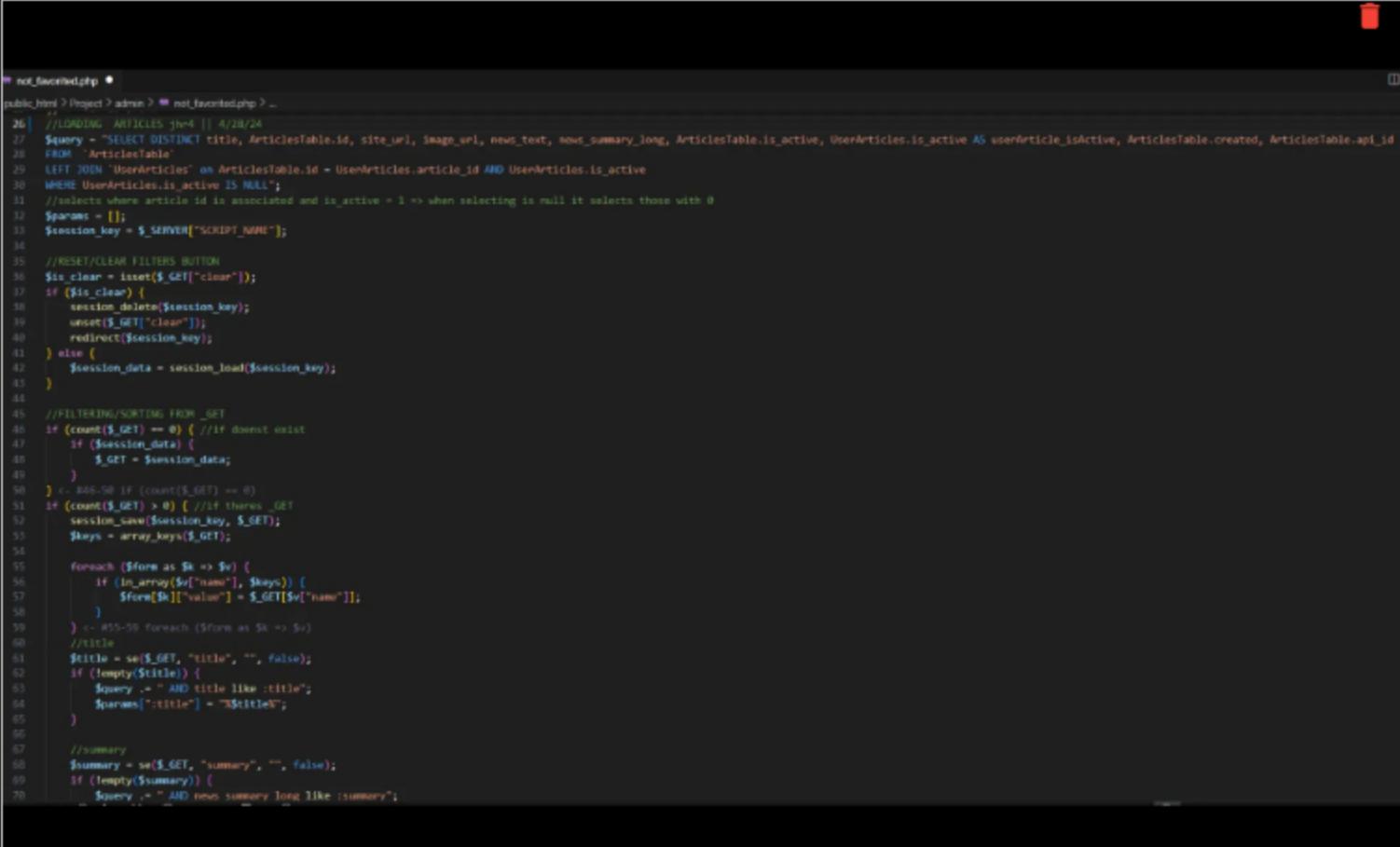
#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the code related to fetching all unassociated entities (including the query)
<input checked="" type="checkbox"/> #2	1	Show the code related to the display of the results
<input checked="" type="checkbox"/> #3	1	Each record should have a button/link for single view
<input checked="" type="checkbox"/> #4	1	Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)
<input checked="" type="checkbox"/> #5	1	Show the logic related to the count of the items on the page (this value should change based on the filter applied)
<input checked="" type="checkbox"/> #6	1	Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

#7	1	Include ucid/date comments for each code screenshot
#8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```

not_favorited.php •
public_html > Project > admin > not_favorited.php > ...
26 //LOADING ARTICLES jhrd || 4/28/24
27 $query = "SELECT DISTINCT title, ArticlesTable.id, site_url, image_url, news_text, news_summary_long, ArticlesTable.is_active, UserArticles.is_active AS userArticle_isActive, ArticlesTable.created, ArticlesTable.api_id
28 FROM `ArticlesTable`
29 LEFT JOIN `UserArticles` ON ArticlesTable.id = UserArticles.article_id AND UserArticles.is_active
30 WHERE UserArticles.is_active IS NULL";
31 //selects where article id is associated and is_active = 1 => when selecting is null it selects those with 0
32 $params = [];
33 $session_key = $_SERVER["SCRIPT_NAME"];
34
35 //RESET/CLEAR FILTERS BUTTON
36 $is_clear = isset($_GET["clear"]);
37 if ($is_clear) {
38     session_delete($session_key);
39     unset($_GET["clear"]);
40     redirect($session_key);
41 } else {
42     $session_data = session_load($session_key);
43 }
44
45 //FILTERING/SORTING FROM _GET
46 if (count($_GET) == 0) { //if doesn't exist
47     if ($session_data) {
48         $_GET = $session_data;
49     }
50 } <- #46-50 IF (count($_GET) == 0)
51 if (count($_GET) > 0) { //if there's _GET
52     session_save($session_key, $_GET);
53     $keys = array_keys($_GET);
54
55     foreach ($form as $k => $v) {
56         if (!in_array($v["name"], $keys)) {
57             $form[$k]["value"] = $_GET[$v["name"]];
58         }
59     } <- #55-59 foreach ($form as $k => $v)
60     /title
61     $title = se($_GET, "title", "", false);
62     if (empty($title)) {
63         $query .= " AND title like :title";
64         $params[":title"] = "%$title%";
65     }
66
67     //summary
68     $summary = se($_GET, "summary", "", false);
69     if (empty($summary)) {
70         $query .= " AND news_summary_long like :summary";
71     }

```

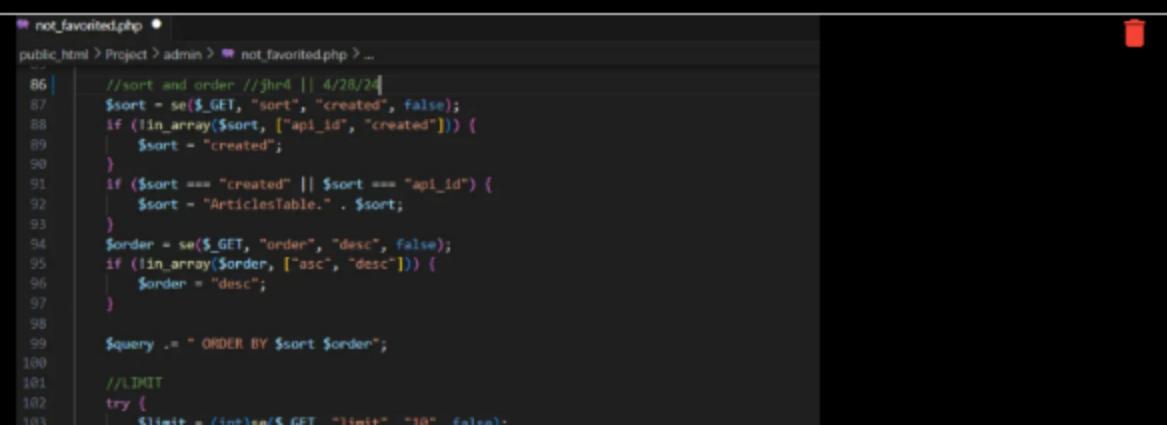
Fetching non associated users query on top. Some Filter/sorting logic below. UCID/Date on Top.

Checklist Items (3)

#1 Show the code related to fetching all unassociated entities (including the query)

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots



```

not_favorited.php •
public_html > Project > admin > not_favorited.php > ...
86 //sort and order //jhrd || 4/28/24
87 $sort = se($_GET, "sort", "created", false);
88 if (!in_array($sort, ["api_id", "created"])) {
89     $sort = "created";
90 }
91 if ($sort === "created" || $sort === "api_id") {
92     $sort = "ArticlesTable." . $sort;
93 }
94 $order = se($_GET, "order", "desc", false);
95 if (!in_array($order, ["asc", "desc"])) {
96     $order = "desc";
97 }
98
99 $query .= " ORDER BY $sort $order";
100
101 //LIMIT
102 try {
103     $limit = (int)se($_GET, "limit", "10", false);
104     $offset = (int)se($_GET, "offset", "0", false);
105     $query .= " LIMIT $limit OFFSET $offset";
106 }
107 catch (Exception $e) {
108     echo "Error: " . $e->getMessage();
109 }
110
111 //EXECUTE
112 $stmt = $pdo->prepare($query);
113 $stmt->execute();
114
115 //GET RESULTS
116 $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
117
118 //PAGINATION
119 $total_results = count($results);
120 $total_pages = ceil($total_results / $limit);
121
122 //PAGINATION LOGIC
123 $current_page = 1;
124 if (isset($_GET['page'])) {
125     $current_page = (int) $_GET['page'];
126 }
127
128 //PAGINATION HTML
129 $pagination = '';
130 for ($i = 1; $i <= $total_pages; $i++) {
131     $class = ($i == $current_page) ? 'active' : '';
132     $pagination .= "\$i ";
133 }
134
135 //DISPLAY RESULTS
136 echo $results;
137
138 //FREE STATEMENT
139 $stmt->close();
140
141 //END
142
143 //SESSION
144 if ($session_key) {
145     session_update($session_key);
146 }
147
148 //REDIRECT
149 redirect($session_key);
150
151 //END
152
153 //SESSION
154 if ($session_key) {
155     session_destroy($session_key);
156 }
157
158 //REDIRECT
159 redirect($session_key);
160
161 //END
162
163 //SESSION
164 if ($session_key) {
165     session_start($session_key);
166 }
167
168 //REDIRECT
169 redirect($session_key);
170
171 //END
172
173 //SESSION
174 if ($session_key) {
175     session_end($session_key);
176 }
177
178 //REDIRECT
179 redirect($session_key);
180
181 //END
182
183 //SESSION
184 if ($session_key) {
185     session_unset($session_key);
186 }
187
188 //REDIRECT
189 redirect($session_key);
190
191 //END
192
193 //SESSION
194 if ($session_key) {
195     session_destroy($session_key);
196 }
197
198 //REDIRECT
199 redirect($session_key);
200
201 //END
202
203 //SESSION
204 if ($session_key) {
205     session_start($session_key);
206 }
207
208 //REDIRECT
209 redirect($session_key);
210
211 //END
212
213 //SESSION
214 if ($session_key) {
215     session_end($session_key);
216 }
217
218 //REDIRECT
219 redirect($session_key);
220
221 //END
222
223 //SESSION
224 if ($session_key) {
225     session_unset($session_key);
226 }
227
228 //REDIRECT
229 redirect($session_key);
230
231 //END
232
233 //SESSION
234 if ($session_key) {
235     session_destroy($session_key);
236 }
237
238 //REDIRECT
239 redirect($session_key);
240
241 //END
242
243 //SESSION
244 if ($session_key) {
245     session_start($session_key);
246 }
247
248 //REDIRECT
249 redirect($session_key);
250
251 //END
252
253 //SESSION
254 if ($session_key) {
255     session_end($session_key);
256 }
257
258 //REDIRECT
259 redirect($session_key);
260
261 //END
262
263 //SESSION
264 if ($session_key) {
265     session_unset($session_key);
266 }
267
268 //REDIRECT
269 redirect($session_key);
270
271 //END
272
273 //SESSION
274 if ($session_key) {
275     session_destroy($session_key);
276 }
277
278 //REDIRECT
279 redirect($session_key);
280
281 //END
282
283 //SESSION
284 if ($session_key) {
285     session_start($session_key);
286 }
287
288 //REDIRECT
289 redirect($session_key);
290
291 //END
292
293 //SESSION
294 if ($session_key) {
295     session_end($session_key);
296 }
297
298 //REDIRECT
299 redirect($session_key);
300
301 //END
302
303 //SESSION
304 if ($session_key) {
305     session_unset($session_key);
306 }
307
308 //REDIRECT
309 redirect($session_key);
310
311 //END
312
313 //SESSION
314 if ($session_key) {
315     session_destroy($session_key);
316 }
317
318 //REDIRECT
319 redirect($session_key);
320
321 //END
322
323 //SESSION
324 if ($session_key) {
325     session_start($session_key);
326 }
327
328 //REDIRECT
329 redirect($session_key);
330
331 //END
332
333 //SESSION
334 if ($session_key) {
335     session_end($session_key);
336 }
337
338 //REDIRECT
339 redirect($session_key);
340
341 //END
342
343 //SESSION
344 if ($session_key) {
345     session_unset($session_key);
346 }
347
348 //REDIRECT
349 redirect($session_key);
350
351 //END
352
353 //SESSION
354 if ($session_key) {
355     session_destroy($session_key);
356 }
357
358 //REDIRECT
359 redirect($session_key);
360
361 //END
362
363 //SESSION
364 if ($session_key) {
365     session_start($session_key);
366 }
367
368 //REDIRECT
369 redirect($session_key);
370
371 //END
372
373 //SESSION
374 if ($session_key) {
375     session_end($session_key);
376 }
377
378 //REDIRECT
379 redirect($session_key);
380
381 //END
382
383 //SESSION
384 if ($session_key) {
385     session_unset($session_key);
386 }
387
388 //REDIRECT
389 redirect($session_key);
390
391 //END
392
393 //SESSION
394 if ($session_key) {
395     session_destroy($session_key);
396 }
397
398 //REDIRECT
399 redirect($session_key);
400
401 //END
402
403 //SESSION
404 if ($session_key) {
405     session_start($session_key);
406 }
407
408 //REDIRECT
409 redirect($session_key);
410
411 //END
412
413 //SESSION
414 if ($session_key) {
415     session_end($session_key);
416 }
417
418 //REDIRECT
419 redirect($session_key);
420
421 //END
422
423 //SESSION
424 if ($session_key) {
425     session_unset($session_key);
426 }
427
428 //REDIRECT
429 redirect($session_key);
430
431 //END
432
433 //SESSION
434 if ($session_key) {
435     session_destroy($session_key);
436 }
437
438 //REDIRECT
439 redirect($session_key);
440
441 //END
442
443 //SESSION
444 if ($session_key) {
445     session_start($session_key);
446 }
447
448 //REDIRECT
449 redirect($session_key);
450
451 //END
452
453 //SESSION
454 if ($session_key) {
455     session_end($session_key);
456 }
457
458 //REDIRECT
459 redirect($session_key);
460
461 //END
462
463 //SESSION
464 if ($session_key) {
465     session_unset($session_key);
466 }
467
468 //REDIRECT
469 redirect($session_key);
470
471 //END
472
473 //SESSION
474 if ($session_key) {
475     session_destroy($session_key);
476 }
477
478 //REDIRECT
479 redirect($session_key);
480
481 //END
482
483 //SESSION
484 if ($session_key) {
485     session_start($session_key);
486 }
487
488 //REDIRECT
489 redirect($session_key);
490
491 //END
492
493 //SESSION
494 if ($session_key) {
495     session_end($session_key);
496 }
497
498 //REDIRECT
499 redirect($session_key);
500
501 //END
502
503 //SESSION
504 if ($session_key) {
505     session_unset($session_key);
506 }
507
508 //REDIRECT
509 redirect($session_key);
510
511 //END
512
513 //SESSION
514 if ($session_key) {
515     session_destroy($session_key);
516 }
517
518 //REDIRECT
519 redirect($session_key);
520
521 //END
522
523 //SESSION
524 if ($session_key) {
525     session_start($session_key);
526 }
527
528 //REDIRECT
529 redirect($session_key);
530
531 //END
532
533 //SESSION
534 if ($session_key) {
535     session_end($session_key);
536 }
537
538 //REDIRECT
539 redirect($session_key);
540
541 //END
542
543 //SESSION
544 if ($session_key) {
545     session_unset($session_key);
546 }
547
548 //REDIRECT
549 redirect($session_key);
550
551 //END
552
553 //SESSION
554 if ($session_key) {
555     session_destroy($session_key);
556 }
557
558 //REDIRECT
559 redirect($session_key);
560
561 //END
562
563 //SESSION
564 if ($session_key) {
565     session_start($session_key);
566 }
567
568 //REDIRECT
569 redirect($session_key);
570
571 //END
572
573 //SESSION
574 if ($session_key) {
575     session_end($session_key);
576 }
577
578 //REDIRECT
579 redirect($session_key);
580
581 //END
582
583 //SESSION
584 if ($session_key) {
585     session_unset($session_key);
586 }
587
588 //REDIRECT
589 redirect($session_key);
590
591 //END
592
593 //SESSION
594 if ($session_key) {
595     session_destroy($session_key);
596 }
597
598 //REDIRECT
599 redirect($session_key);
600
601 //END
602
603 //SESSION
604 if ($session_key) {
605     session_start($session_key);
606 }
607
608 //REDIRECT
609 redirect($session_key);
610
611 //END
612
613 //SESSION
614 if ($session_key) {
615     session_end($session_key);
616 }
617
618 //REDIRECT
619 redirect($session_key);
620
621 //END
622
623 //SESSION
624 if ($session_key) {
625     session_unset($session_key);
626 }
627
628 //REDIRECT
629 redirect($session_key);
630
631 //END
632
633 //SESSION
634 if ($session_key) {
635     session_destroy($session_key);
636 }
637
638 //REDIRECT
639 redirect($session_key);
640
641 //END
642
643 //SESSION
644 if ($session_key) {
645     session_start($session_key);
646 }
647
648 //REDIRECT
649 redirect($session_key);
650
651 //END
652
653 //SESSION
654 if ($session_key) {
655     session_end($session_key);
656 }
657
658 //REDIRECT
659 redirect($session_key);
660
661 //END
662
663 //SESSION
664 if ($session_key) {
665     session_unset($session_key);
666 }
667
668 //REDIRECT
669 redirect($session_key);
670
671 //END
672
673 //SESSION
674 if ($session_key) {
675     session_destroy($session_key);
676 }
677
678 //REDIRECT
679 redirect($session_key);
680
681 //END
682
683 //SESSION
684 if ($session_key) {
685     session_start($session_key);
686 }
687
688 //REDIRECT
689 redirect($session_key);
690
691 //END
692
693 //SESSION
694 if ($session_key) {
695     session_end($session_key);
696 }
697
698 //REDIRECT
699 redirect($session_key);
700
701 //END
702
703 //SESSION
704 if ($session_key) {
705     session_unset($session_key);
706 }
707
708 //REDIRECT
709 redirect($session_key);
710
711 //END
712
713 //SESSION
714 if ($session_key) {
715     session_destroy($session_key);
716 }
717
718 //REDIRECT
719 redirect($session_key);
720
721 //END
722
723 //SESSION
724 if ($session_key) {
725     session_start($session_key);
726 }
727
728 //REDIRECT
729 redirect($session_key);
730
731 //END
732
733 //SESSION
734 if ($session_key) {
735     session_end($session_key);
736 }
737
738 //REDIRECT
739 redirect($session_key);
740
741 //END
742
743 //SESSION
744 if ($session_key) {
745     session_unset($session_key);
746 }
747
748 //REDIRECT
749 redirect($session_key);
750
751 //END
752
753 //SESSION
754 if ($session_key) {
755     session_destroy($session_key);
756 }
757
758 //REDIRECT
759 redirect($session_key);
760
761 //END
762
763 //SESSION
764 if ($session_key) {
765     session_start($session_key);
766 }
767
768 //REDIRECT
769 redirect($session_key);
770
771 //END
772
773 //SESSION
774 if ($session_key) {
775     session_end($session_key);
776 }
777
778 //REDIRECT
779 redirect($session_key);
780
781 //END
782
783 //SESSION
784 if ($session_key) {
785     session_unset($session_key);
786 }
787
788 //REDIRECT
789 redirect($session_key);
790
791 //END
792
793 //SESSION
794 if ($session_key) {
795     session_destroy($session_key);
796 }
797
798 //REDIRECT
799 redirect($session_key);
800
801 //END
802
803 //SESSION
804 if ($session_key) {
805     session_start($session_key);
806 }
807
808 //REDIRECT
809 redirect($session_key);
810
811 //END
812
813 //SESSION
814 if ($session_key) {
815     session_end($session_key);
816 }
817
818 //REDIRECT
819 redirect($session_key);
820
821 //END
822
823 //SESSION
824 if ($session_key) {
825     session_unset($session_key);
826 }
827
828 //REDIRECT
829 redirect($session_key);
830
831 //END
832
833 //SESSION
834 if ($session_key) {
835     session_destroy($session_key);
836 }
837
838 //REDIRECT
839 redirect($session_key);
840
841 //END
842
843 //SESSION
844 if ($session_key) {
845     session_start($session_key);
846 }
847
848 //REDIRECT
849 redirect($session_key);
850
851 //END
852
853 //SESSION
854 if ($session_key) {
855     session_end($session_key);
856 }
857
858 //REDIRECT
859 redirect($session_key);
860
861 //END
862
863 //SESSION
864 if ($session_key) {
865     session_unset($session_key);
866 }
867
868 //REDIRECT
869 redirect($session_key);
870
871 //END
872
873 //SESSION
874 if ($session_key) {
875     session_destroy($session_key);
876 }
877
878 //REDIRECT
879 redirect($session_key);
880
881 //END
882
883 //SESSION
884 if ($session_key) {
885     session_start($session_key);
886 }
887
888 //REDIRECT
889 redirect($session_key);
890
891 //END
892
893 //SESSION
894 if ($session_key) {
895     session_end($session_key);
896 }
897
898 //REDIRECT
899 redirect($session_key);
900
901 //END
902
903 //SESSION
904 if ($session_key) {
905     session_unset($session_key);
906 }
907
908 //REDIRECT
909 redirect($session_key);
910
911 //END
912
913 //SESSION
914 if ($session_key) {
915     session_destroy($session_key);
916 }
917
918 //REDIRECT
919 redirect($session_key);
920
921 //END
922
923 //SESSION
924 if ($session_key) {
925     session_start($session_key);
926 }
927
928 //REDIRECT
929 redirect($session_key);
930
931 //END
932
933 //SESSION
934 if ($session_key) {
935     session_end($session_key);
936 }
937
938 //REDIRECT
939 redirect($session_key);
940
941 //END
942
943 //SESSION
944 if ($session_key) {
945     session_unset($session_key);
946 }
947
948 //REDIRECT
949 redirect($session_key);
950
951 //END
952
953 //SESSION
954 if ($session_key) {
955     session_destroy($session_key);
956 }
957
958 //REDIRECT
959 redirect($session_key);
960
961 //END
962
963 //SESSION
964 if ($session_key) {
965     session_start($session_key);
966 }
967
968 //REDIRECT
969 redirect($session_key);
970
971 //END
972
973 //SESSION
974 if ($session_key) {
975     session_end($session_key);
976 }
977
978 //REDIRECT
979 redirect($session_key);
980
981 //END
982
983 //SESSION
984 if ($session_key) {
985     session_unset($session_key);
986 }
987
988 //REDIRECT
989 redirect($session_key);
990
991 //END
992
993 //SESSION
994 if ($session_key) {
995     session_destroy($session_key);
996 }
997
998 //REDIRECT
999 redirect($session_key);
1000
1001 //END
1002
1003 //SESSION
1004 if ($session_key) {
1005     session_start($session_key);
1006 }
1007
1008 //REDIRECT
1009 redirect($session_key);
1010
1011 //END
1012
1013 //SESSION
1014 if ($session_key) {
1015     session_end($session_key);
1016 }
1017
1018 //REDIRECT
1019 redirect($session_key);
1020
1021 //END
1022
1023 //SESSION
1024 if ($session_key) {
1025     session_unset($session_key);
1026 }
1027
1028 //REDIRECT
1029 redirect($session_key);
1030
1031 //END
1032
1033 //SESSION
1034 if ($session_key) {
1035     session_destroy($session_key);
1036 }
1037
1038 //REDIRECT
1039 redirect($session_key);
1040
1041 //END
1042
1043 //SESSION
1044 if ($session_key) {
1045     session_start($session_key);
1046 }
1047
1048 //REDIRECT
1049 redirect($session_key);
1050
1051 //END
1052
1053 //SESSION
1054 if ($session_key) {
1055     session_end($session_key);
1056 }
1057
1058 //REDIRECT
1059 redirect($session_key);
1060
1061 //END
1062
1063 //SESSION
1064 if ($session_key) {
1065     session_unset($session_key);
1066 }
1067
1068 //REDIRECT
1069 redirect($session_key);
1070
1071 //END
1072
1073 //SESSION
1074 if ($session_key) {
1075     session_destroy($session_key);
1076 }
1077
1078 //REDIRECT
1079 redirect($session_key);
1080
1081 //END
1082
1083 //SESSION
1084 if ($session_key) {
1085     session_start($session_key);
1086 }
1087
1088 //REDIRECT
1089 redirect($session_key);
1090
1091 //END
1092
1093 //SESSION
1094 if ($session_key) {
1095     session_end($session_key);
1096 }
1097
1098 //REDIRECT
1099 redirect($session_key);
1100
1101 //END
1102
1103 //SESSION
1104 if ($session_key) {
1105     session_unset($session_key);
1106 }
1107
1108 //REDIRECT
1109 redirect($session_key);
1110
1111 //END
1112
1113 //SESSION
1114 if ($session_key) {
1115     session_destroy($session_key);
1116 }
1117
1118 //REDIRECT
1119 redirect($session_key);
1120
1121 //END
1122
1123 //SESSION
1124 if ($session_key) {
1125     session_start($session_key);
1126 }
1127
1128 //REDIRECT
1129 redirect($session_key);
1130
1131 //END
1132
1133 //SESSION
1134 if ($session_key) {
1135     session_end($session_key);
1136 }
1137
1138 //REDIRECT
1139 redirect($session_key);
1140
1141 //END
1142
1143 //SESSION
1144 if ($session_key) {
1145     session_unset($session_key);
1146 }
1147
1148 //REDIRECT
1149 redirect($session_key);
1150
1151 //END
1152
1153 //SESSION
1154 if ($session_key) {
1155     session_destroy($session_key);
1156 }
1157
1158 //REDIRECT
1159 redirect($session_key);
1160
1161 //END
1162
1163 //SESSION
1164 if ($session_key) {
1165     session_start($session_key);
1166 }
1167
1168 //REDIRECT
1169 redirect($session_key);
1170
1171 //END
1172
1173 //SESSION
1174 if ($session_key) {
1175     session_end($session_key);
1176 }
1177
1178 //REDIRECT
1179 redirect($session_key);
1180
1181 //END
1182
1183 //SESSION
1184 if ($session_key) {
1185     session_unset($session_key);
1186 }
1187
1188 //REDIRECT
1189 redirect($session_key);
1190
1191 //END
1192
1193 //SESSION
1194 if ($session_key) {
1195     session_destroy($session_key);
1196 }
1197
1198 //REDIRECT
1199 redirect($session_key);
1200
1201 //END
1202
1203 //SESSION
1204 if ($session_key) {
1205     session_start($session_key);
1206 }
1207
1208 //REDIRECT
1209 redirect($session_key);
1210
1211 //END
1212
1213 //SESSION
1214 if ($session_key) {
1215     session_end($session_key);
1216 }
1217
1218 //REDIRECT
1219 redirect($session_key);
1220
1221 //END
1222
1223 //SESSION
1224 if ($session_key) {
1225     session_unset($session_key);
1226 }
1227
1228 //REDIRECT
1229 redirect($session_key);
1230
1231 //END
1232
1233 //SESSION
1234 if ($session_key) {
1235     session_destroy($session_key);
1236 }
1237
1238 //REDIRECT
1239 redirect($session_key);
1240
1241 //END
1242
1243 //SESSION
1244 if ($session_key) {
1245     session_start($session_key);
1246 }
1247
1248 //REDIRECT
1249 redirect($session_key);
1250
1251 //END
1252
1253 //SESSION
1254 if ($session_key) {
1255     session_end($session_key);
1256 }
1257
1258 //REDIRECT
1259 redirect($session_key);
1260
1261 //END
1262
1263 //SESSION
1264 if ($session_key) {
1265     session_unset($session_key);
1266 }
1267
1268 //REDIRECT
1269 redirect($session_key);
1270
1271 //END
1272
1273 //SESSION
1274 if ($session_key) {
1275     session_destroy($session_key);
1276 }
1277
1278 //REDIRECT
1279 redirect($session_key);
1280
1281 //END
1282
1283 //SESSION
1284 if ($session_key) {
1285     session_start($session_key);
1286 }
1287
1288 //REDIRECT
1289 redirect($session_key);
1290
1291 //END
1292
1293 //SESSION
1294 if ($session_key) {
1295     session_end($session_key);
1296 }
1297
1298 //REDIRECT
1299 redirect($session_key);
1300
1301 //END
1302
1303 //SESSION
1304 if ($session_key) {
1305     session_unset($session_key);
1306 }
1307
1308 //REDIRECT
1309 redirect($session_key);
1310
1311 //END
1312
1313 //SESSION
1314 if ($session_key) {
1315     session_destroy($session_key);
1316 }
1317
1318 //REDIRECT
1319 redirect($session_key);
1320
1321 //END
1322
1323 //SESSION
1324 if ($session_key) {
1325     session_start($session_key);
1326 }
1327
1328 //REDIRECT
1329 redirect($session_key);
1330
1331 //END
1332
1333 //SESSION
1334 if ($session_key) {
1335     session_end($session_key);
1336 }
1337
1338 //REDIRECT
1339 redirect($session_key);
1340
1341 //END
1342
1343 //SESSION
1344 if ($session_key) {
1345     session_unset($session_key);
1346 }
1347
1348 //REDIRECT
1349 redirect($session_key);
1350
1351 //END
1352
1353 //SESSION
1354 if ($session_key) {
1355     session_destroy($session_key);
1356 }
1357
1358 //REDIRECT
1359 redirect($session_key);
1360
1361 //END
1362
1363 //SESSION
1364 if ($session_key) {
1365     session_start($session_key);
1366 }
1367
1368 //REDIRECT
1369 redirect($session_key);
1370
1371 //END
1372
1373 //SESSION
1374 if ($session_key) {
1375     session_end($session_key);
1376 }
1377
1378 //REDIRECT
1379 redirect($session_key);
1380
1381 //END
1382
1383 //SESSION
1384 if ($session_key) {
1385     session_unset($session_key);
1386 }
1387
1388 //REDIRECT
1389 redirect($session_key);
1390
1391 //END
1392
1393 //SESSION
1394 if ($session_key) {
1395     session_destroy($session_key);
1396 }
1397
1398 //REDIRECT
1399 redirect($session_key);
1400
1401 //END
1402
1403 //SESSION
1404 if ($session_key) {
1405     session_start($session_key);
1406 }
1407
1408 //REDIRECT
1409 redirect($session_key);
1410
1411 //END
1412
1413 //SESSION
1414 if ($session_key) {
1415     session_end($session_key);
1416 }
1417
1418 //REDIRECT
1419 redirect($session_key);
1420
1421 //END
1422
1423 //SESSION
1424 if ($session_key) {
1425     session_unset($session_key);
1426 }
1427
1428 //REDIRECT
1429 redirect($session_key);
1430
1431 //END
1432
1433 //SESSION
1434 if ($session_key) {
1435     session_destroy($session_key);
1436 }
1437
1438 //REDIRECT
1439 redirect($session_key);
1440
1441 //END
1442
1443 //SESSION
1444 if ($session_key) {
1445     session_start($session_key);
1446 }
1447
1448 //REDIRECT
1449 redirect($session_key);
1450
1451 //END
1452
1453 //SESSION
1454 if ($session_key) {
1455     session_end($session_key);
1456 }
1457
1458 //REDIRECT
1459 redirect($session_key);
1460
1461 //END
1462
1463 //SESSION
1464 if ($session_key) {
1465     session_unset($session_key);
1466 }
1467
1468 //REDIRECT
1469 redirect($session_key);
1470
1471 //END
1472
1473 //SESSION
1474 if ($session_key) {
1475     session_destroy($session_key);
1476 }
1477
1478 //REDIRECT
1479 redirect($session_key);
1480
1481 //END
1482
1483 //SESSION
1484 if ($session_key) {
1485     session_start($session_key);
1486 }
1487
1488 //REDIRECT
1489 redirect($session_key);
1490
1491 //END
1492
1493 //SESSION
1494 if ($session_key) {
1495     session_end($session_key);
1496 }
1497
1498 //REDIRECT
1499 redirect($session_key);
1500
1501 //END
1502
1503 //SESSION
1504 if ($session_key) {
1505     session_unset($session_key);
1506 }
1507
1508 //REDIRECT
1509 redirect($session_key);
1510
1511 //END
1512
1513 //SESSION
1514 if ($session_key) {
1515     session_destroy($session_key);
1516 }
1517
1518 //REDIRECT
1519 redirect($session_key);
1520
1521 //END
1522
1523 //SESSION
1524 if ($session_key) {
1525     session_start($session_key);
1526 }
1527
1528 //REDIRECT
1529 redirect($session_key);
1530
1531 //END
1532
1533 //SESSION
1534 if ($session_key) {
1535     session_end($session_key);
1536 }
1537
1538 //REDIRECT
1539 redirect($session_key);
1540
1541 //END
1542
1543 //SESSION
1544 if ($session_key) {
1545     session_unset($session_key);
1546 }
1547
1548 //REDIRECT
1549 redirect($session_key);
1550
1551 //END
1552
1553 //SESSION
1554 if ($session_key) {
1555     session_destroy($session_key);
1556 }
1557
1558 //REDIRECT
1559 redirect($session_key);
1560
1561 //END
1562
1563 //SESSION
1564 if ($session_key) {
1565     session_start($session_key);
1566 }
1567
1568 //REDIRECT
1569 redirect($session_key);
1570
1571 //END
1572
1573 //SESSION
1574 if ($session_key) {
1
```

```

103     $query .= " ORDER BY $sort $order";
104 }
105 catch (PDOException $error) {
106     $limit = 10;
107 }
108 if ($limit < 1 || $limit > 100) {
109     $limit = 10;
110 }
111 $query .= " LIMIT $limit";
112 } else { //If No Session data loaded && no filters -> user will see leastest articles first
113     $sort = "ArticlesTable.created";
114     $order = "desc";
115     $query .= " ORDER BY $sort $order";
116 }
117 $db = getDB();
118 $stmt = $db->prepare($query);
119 $results = [];
120 try {
121     $stmt->execute($params);
122     $r = $stmt->fetchAll();
123     if ($r) {
124         $results = $r;
125     }
126 } catch (PDOException $error) {
127     error_log("Error fetching articles: " . var_export($error, true));
128     Flash("An error occurred", "danger");
129 }

```

Filter Logic for Limit 1-100 in middle. UCID / Date on Top.

Checklist Items (1)

#6 Show the logic related to filter/sort (should include a partial match for username) (limit should be constrained to 1-100 otherwise default to 10)

Displaying Results code. Button to single view on bottom. UCID / Date on top.

Checklist Items (2)

#2 Show the code related to the display of the results

#3 Each record should have a button/link for single view

Logic for total (Top Left + Right) and shown count (Top Left). UCID/Date on top for both.

Checklist Items (2)

#4 Show the logic related to the count of all unassociated items (even the ones not shown in the filtered results)

#5 Show the logic related to the count of the items on the page (this value should change based on the filter applied)

Task #3 - Points: 1

Text: Explain the solution

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Mention how you determine the result list (include the unassociated logic and filters)
<input checked="" type="checkbox"/> #2	1	Mention the logic behind the two counts (total items and visible result count)

Response:

For this page as it just shows what's not associated with any user it was as simple as selecting all the data I want from the Articles table and left joining on the UserArticles where the article id is the same as the article association id and the association is active (1/true). This seems contradictory, but I then add a where clause that limits it to where is_active IS NULL. Therefore, all the articles that aren't associated are shown (the values that are null upon left joining). The distinct doesn't really impact it.

For the shown count I do the same thing again, just counting the entities/articles in the results array from the query utilizing the count() method.

For the total count I just copy the same original query without any filters and it does the same counting of the entities.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with /pull/#. Same pull request shouldn't be used for each feature

URL #1

<https://github.com/Jhr-4/jhr4-IT202-008/pull/65>

URL #2

https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/Project/admin/not_favorited.php

Admin Association Management (like UserRoles) (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshots of the page

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Show the search form with valid data
<input checked="" type="checkbox"/> #2	1	Show the results of the search
<input checked="" type="checkbox"/> #3	1	Show the result of entities and users being associated and unassociated
<input checked="" type="checkbox"/> #4	1	Make sure the heroku dev url is visible in the address bar
<input checked="" type="checkbox"/> #5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

The screenshot shows a web application interface for managing user associations. At the top, there's a navigation bar with links for 'Home', 'Profile', 'My Favorites', 'Manage Roles', 'Manage Articles', 'Manage Favorites', and 'Logout'. Below the navigation, a search bar is labeled 'FILTER USERS: Username' with a placeholder 'Username'. The main content area is titled 'Assign Favorites' and contains a table with columns for 'User' and 'Favorite'. The 'User' column lists several names, each preceded by a blue checkbox. The 'Favorite' column shows a list of articles or items. A large red 'X' icon is positioned in the top right corner of the screenshot area.

The screenshot shows a search interface with two search bars at the top. The left search bar contains 'hello' and the right one contains 'nasa'. Below the search bars are two buttons: 'Filter' and 'Reset'. A red button labeled 'Toggle Favorites' is located below the search bars. The main area is divided into two sections: 'Users [Article ID]' and 'Articles to Assign [(ID) Title]'. The 'Users' section lists three users: 'hello' (30 active, 300 inactive), 'hello2' (30 active, 300 inactive), and 'hello3' (30 active, 37 inactive). The 'Articles' section lists several articles, with the first few being: '(30) Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars ■', '(32) Investing in Space: NASA's tool for seeing the solar eclipse ■', '(37) NASA asks the commercial space industry for a rugged, long-lived lunar rover ■', '(42) NASA Picks Three Companies for Lunar Terrain Vehicle Feasibility Studies ■', '(43) NASA selects three companies to advance Artemis lunar rover designs ■', '(275) NASA to Develop a Framework for Space Sustainability ■', '(276) NASA rolls out new space sustainability strategy ■', '(300) NASA Invites Media to Mars Sample Return Update ■', '(324) NASA shares updated render of the Cargo Starship variant ■', '(327) NASA planning September launch of Mars smallsat mission on first New Glenn ■', and '(337) Electron launches South Korean imaging satellite and NASA solar sail ■'.

Valid data in search form (with results shown below) Dev URL on top

Checklist Items (4)

#1 Show the search form with valid data

#2 Show the results of the search

#4 Make sure the heroku dev url is visible in the address bar

#5 Clearly caption screenshots

The screenshot shows the 'Assign Favorites' page. At the top, there are two search bars: 'FILTER USERS: Username' containing 'hello' and 'FILTER ARTICLES: Article Title' containing 'nasa'. Below these are 'Filter' and 'Reset' buttons, and a 'Toggle Favorites' button. The main area is divided into 'Users [Article ID]' and 'Articles to Assign [(ID) Title]'. The 'Users' section lists three users: 'hello' (30 active, 300 inactive, 337 active), 'hello2' (30 active, 300 inactive, 337 active), and 'hello3' (30 active, 37 inactive, 337 active). The 'Articles' section lists the same set of articles as the previous screenshot, including '(30) Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars ■' through '(337) Electron launches South Korean imaging satellite and NASA solar sail ■'.

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the 'Assign Favorites' interface. At the top, there are two search boxes: 'FILTER USERS: Username' containing 'hello' and 'FILTER ARTICLES: Article Title' containing 'nasa'. Below these are two tables. The left table, 'Users [Article ID]', lists three users: 'hello' (inactive), '300 (inactive)', '337 (active); 'hello2' (inactive), '300 (inactive)', '337 (active); and 'hello3' (inactive), '37 (inactive)', '337 (active)'. The right table, 'Articles to Assign [(ID) Title]', lists several articles, including '(30) Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars' through '(327) NASA planning September launch of Mars smallsat mission on first New Glenn'. A success message 'Updated Article Favorites' is displayed at the top of the page.

Article ID 337 unassociated from all uses.

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the 'Assign Favorites' interface. The setup is identical to the previous one: 'hello' in the user filter and 'nasa' in the article filter. However, the results are different. The left table, 'Users [Article ID]', now displays 'No Filter on Users Set Yet.' The right table, 'Articles to Assign [(ID) Title]', shows a subset of the previous list, specifically '(30) Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars' through '(35) Kratos demonstrates satellite internet tech for U.S. Army'. This indicates that while the filters were applied, no actual associations were made.

(38) Calling all space stewards: could direct-to-smartphone elevate space sustainability? ■
(39) US Agency Selects ICEYE to Provide Flood Impact Data ■
(40) Hydrosat wins NOAA contract ■
(41) Scout Space eyes military customers for space domain awareness sensor ■
(42) NASA Picks Three Companies for Lunar Terrain Vehicle Feasibility Studies ■
(43) NASA selects three companies to advance Artemis lunar rover designs ■
(62) Here are the winners and losers when it comes to clouds for Mondau's eclipse ■

No Filters Applied YET. search form has valid data entered (NOT SUBMITTED YET)

Checklist Items (1)

#1 Show the search form with valid data

The screenshot shows the 'Assign Favorites' page with two filter sections: 'FILTER USERS: Username' (with 'hello' entered) and 'FILTER ARTICLES: Article Title' (with 'nasa' entered). Below these are two tables: 'Users [Article ID]' and 'Articles to Assign [(ID) Title]'. The 'Users' table lists three users: 'hello' (30 inactive, 300 inactive, 337 active), 'hello2' (30 inactive, 32 active, 300 inactive, 337 active), and 'hello3' (30 inactive, 37 inactive, 337 active). The 'Articles' table lists several NASA-related articles, including titles like 'Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars' and 'NASA invites Media to Mars Sample Return Update'.

Before Association (Selected Article 32, Users hello and hello2)

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

The screenshot shows the 'Assign Favorites' page with the same filters and tables as before. A message 'Updated Article Favorites' is displayed in a modal window. The 'Articles' table now includes a column indicating which users have favorited each article. For example, article 32 is listed under both 'hello' and 'hello2'.

Toggle Favorites

Users [Article ID]		Articles to Assign [(ID) Title]
hello	30 (inactive), 32 (active), 300 (inactive), 337 (active)	(30) Rock Samp led by NASA's Perseverance Embodies Why Rover Came to Mars ■ (32) Investing in Space: NASA's tool for seeing the solar eclipse ■ (37) NASA asks the commercial space industry for a rugged, long-lived lunar rover ■ (42) NASA Picks Three Companies for Lunar Terrain Vehicle Feasibility Studies ■ (43) NASA selects three companies to advance Artemis lunar rover designs ■ (275) NASA to Develop a Framework for Space Sustainability ■ (276) NASA rolls out new space sustainability strategy ■ (300) NASA Invites Media to Mars Sample Return Update ■ (324) NASA shares updated render of the Cargo Starship variant ■ (327) NASA planning September launch of Mars smallsat mission on first New Glenn ■
hello2	30 (inactive), 32 (inactive), 300 (inactive), 337 (active)	
hello3	30 (inactive), 37 (inactive), 337 (active)	
...		

After Association (Updated Article 32, Users hello and hello2)

Checklist Items (1)

#3 Show the result of entities and users being associated and unassociated

Task #2 - Points: 1

Text: Screenshots of the code

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Search form field for finding a partial match of usernames
<input type="checkbox"/> #2	1	Search form field for finding a partial match of entities
<input type="checkbox"/> #3	1	Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)
<input type="checkbox"/> #4	1	Code that generates the checkboxes next to each list (users and entities)
<input type="checkbox"/> #5	1	Code related to submitting the checkbox lists
<input type="checkbox"/> #6	1	Code related to applying the associations upon submission (i.e., add the relationship if it doesn't exist and remove the relationship if it does exist)
<input type="checkbox"/> #7	1	Include ucid/date comments for each code screenshot
<input type="checkbox"/> #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large

```
80 //search for user by username jhrd || 4/28/24
81 $users = [];
82 $username = "";
83 if (isset($_POST["username"])) {
84     $username = se($_POST, "username", "", false);
85     if (!empty($username)) {
86         $db = getDB();
87         $query = "SELECT Users.id, username,
88             (SELECT GROUP_CONCAT(' ', ArticlesTable.id, ' (' , IF(ua.is_active = 1,'active','inactive') ,')) from
89             UserArticles ua JOIN ArticlesTable on ua.article_id = ArticlesTable.id WHERE ua.user_id = Users.id";
90         $params = [];
91
92         if (isset($_POST["title"])) {
93             //title
```

```

94     $title = se($_POST, "title", "", false);
95     if (!empty($title)) {
96         $query .= " AND title like :title";
97         $params[":title"] = "%$title%";
98     }
99 } <- #92-99 if (isset($_POST["title"]))
100 $query .= " LIMIT 25" as articles from Users WHERE username like :username LIMIT 25";
101 $params[":username"] = "%$username%";
102 $stmt = $db->prepare($query);
103 try {
104     $stmt->execute($params);
105     $results = $stmt->fetchAll(PDO::FETCH_ASSOC);
106     if ($results) {
107         $users = $results;
108     }
109 } catch (PDOException $e) {
110     error_log("Error Fetching Users: " . var_export($e, true));
111     flash("An Error Occured while Fetching Users", "danger");
112 }
113 } else {
114     flash("Username must not be empty", "warning");
115 }
116 } <- #85-116 if (!empty($username))
117

```

Code of Username Partial Match Limit 25 (line 100) UCID/Date top

Checklist Items (4)

#1 Search form field for finding a partial match of usernames

#3 Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)

#7 Include ucid/date comments for each code screenshot

#8 Clearly caption screenshots

```

53 //get active articles jhr4 || 4/28/24
54 $active_articles = [];
55 $db = getDB();
56 $query = "SELECT DISTINCT ArticlesTable.id, title FROM `ArticlesTable`"
57 LEFT JOIN `UserArticles` on ArticlesTable.id = UserArticles.article_id WHERE ArticlesTable.is_active = true";
58 $params = [];
59 $title = "";
60 if (isset($_POST["title"])) {
61     //title
62     $title = se($_POST, "title", "", false);
63     if (!empty($title)) {
64         $query .= " AND title like :title LIMIT 25";
65         $params[":title"] = "%$title%";
66     }
67 } <- #60-67 if (isset($_POST["title"]))
68 $stmt = $db->prepare($query);
69 try {
70     $stmt->execute($params);
71     $r = $stmt->fetchAll();
72     if ($r) {
73         $active_articles = $r;
74     }
75 } catch (PDOException $e) {
76     error_log("Error Fetching Articles: " . var_export($e, true));
77     flash("An Error Occured while Fetching Articles", "danger");
78 }
79

```

Code of Articles/Entities Partial Match Limit 25 (line 64) UCID/Date top

Checklist Items (2)**#2 Search form field for finding a partial match of entities****#3 Code related to getting a max of 25 results for each field (i.e., 25 users and 25 entities limit)**

```

154      <table class="table">
155          <tbody>
156              <tr>
157                  <td>
158                      <table class="table"> <!-- jhr4 || 4/28/24 --|
159                          <tr>
160                              <td>
161                                  <table class="table"> <!-- jhr4 || 4/28/24 --|
162                                      <?php if (isset($_POST["username"]) && empty($users)) : ?>
163                                          <p class="text-center mt-3">No Users Found.</p>
164                                      <?php elseif (!isset($_POST["username"]) && empty($users)) : ?>
165                                          <p class="text-center mt-3">No Filter on Users Set Yet.</p>
166                                      <?php endif; ?>
167                                      <?php foreach ($users as $user) : ?>
168                                          <tr>
169                                              <td>
170                                                  <label for="user_<?php se($user, 'id'); ?>"><?php se($user, "username"); ?>
171                                                      <input id="user_<?php se($user, 'id'); ?>" type="checkbox" name="users[]" value="<?php se($user, 'id'); ?>" />
172                                              </label>
173                                          </td>
174                                          <td><?php se($user, "articles", "No Articles"); ?></td>
175                                          </tr>
176                                      <?php endforeach; ?>
177                                  </table>
178                              </td>
179                          <td>
180                              <?php $counter = 0; ?>
181                              <?php if (empty($active_articles)) : ?>
182                                  <p class="text-center mt-3">No Active Articles Found.</p>
183                              <?php endif; ?>
184                              <?php foreach ($active_articles as $article) : ?>
185                                  <?php $counter++; ?>
186                                  <div class="<?php if ($counter % 2 === 0) {>
187                                      se("bg-secondary bg-opacity-25");
188                                  } else {
189                                      se("bg-secondary bg-opacity-50");
190                                  } ?>">
191                                      <label for="article_<?php se($article, 'id'); ?>"><?php se($article, 'id'); ?> <?php se($article, "title"); ?></label>
192                                      <input id="article_<?php se($article, 'id'); ?>" type="checkbox" name="articles[]" value="<?php se($article, 'id'); ?>" />
193                                  </div>
194                              <?php endforeach; ?>
195                          </td>
196                      </tr>
197                  </tbody>
198              </table>
199          </form>
200      
```

Code for checkbox next to the users and next to the articles/entities submitting UCID/Date top**Checklist Items (1)****#4 Code that generates the checkboxes next to each list (users and entities)**

```

11 //attempt to apply associations jhr4 || 4/28/24
12 if (isset($_POST["users"]) && isset($_POST["articles"])) {
13     $user_ids = $_POST["users"];
14     $article_ids = $_POST["articles"];
15     if (empty($user_ids) || empty($article_ids)) {
16         flash("Users and articles should not be empty", "warning");
17     } else {
18         $db = getDB();
19         $query = "INSERT INTO UserArticles (user_id, article_id, is_active) VALUES ";
20         $params = [];
21         $counterUSERS = 0;
22         foreach ($user_ids as $userID) {
23             $counterUSERS++;
24             $counterARTICLES = 0;
25             foreach ($article_ids as $articleID) {
26                 $counterARTICLES++;
27                 $query .= "(:userID:$counterUSERS";
28                 $params[":userID:$counterUSERS"] = $userID;
29                 $query .= ", :articleID:$counterARTICLES";
30                 $params[":articleID:$counterARTICLES"] = $articleID;
31                 $query .= ", 1);";
32             } <- #25-32 foreach ($article_ids as $articleID)
33         } <- #22-33 foreach ($user_ids as $userID)
34         $query = rtrim($query, ",");
35         $query .= " ON DUPLICATE KEY UPDATE is_active = !is_active";
36         /*echo "<pre>";
37         var_dump($query);
38         var_dump($params);
39         echo "</pre>";*/

```

```

40     try {
41         $stmt = $db->prepare($query);
42         $stmt->execute($params);
43         flash("Updated Article Favorites", "success");
44     } catch (PDOException $e) {
45         error_log("Error Applying Association: " . var_export($e, true));
46         flash("An Error Occurred While Toggling Favorites", "danger");
47     }
48 } <- #17-48 else
49 } else if (isset($_POST["users"]) || isset($_POST["articles"])) { //if just one is set
50     flash("Both Users and articles must be selected", "warning");
51 }

```

Code for applying the association. (query adds the relationship on duplicate makes the is_active to not is_active (the opposite of current state) UCID/Date on Top

Checklist Items (2)

#5 Code related to submitting the checkbox lists

#6 Code related to applying the associations upon submission (i.e., add the relationship if it doesn't exist and remove the relationship if it does exist)

The screenshot shows a web page with a form for managing article favorites. At the top is a submit button labeled "Toggle Favorites". Below it is a table with two columns: "Users [Article ID]" and "Articles to Assign [(ID) Title]". The "Users" column contains a table where each user row has a checkbox labeled "user_<?php se(\$user, 'id'); ?>" with the value "user_<?php se(\$user, 'username'); ?>". The "Active Articles" column contains a table where each article row has a checkbox labeled "article_<?php se(\$article, 'id'); ?>" with the value "article_<?php se(\$article, 'title'); ?>". The entire form is submitted via POST method.

```

<form method="POST">
    <button type="submit" class="btn btn-danger mb-3">Toggle Favorites</button><!-- jhr4 || 4/28/24 -->

    <table class="table">
        <thead>
            <th style="width:50dvw;">Users [Article ID]</th>
            <th>Articles to Assign [(ID) Title]</th>
        </thead>
        <tbody>
            <tr>
                <td>
                    <table class="table">
                        <?php if (isset($_POST["username"]) && empty($users)) : ?>
                            <p class="text-center mt-3">No Users Found.</p>
                        <?php elseif (!isset($_POST["username"]) && empty($users)) : ?>
                            <p class="text-center mt-3">No Filter on Users Set Yet.</p>
                        <?php endif; ?>
                        <?php foreach ($users as $user) : ?>
                            <tr>
                                <td>
                                    <label for="user_<?php se($user, 'id'); ?>"><?php se($user, "username"); ?>
                                    <input id="user_<?php se($user, 'id'); ?>" type="checkbox" name="users[]" value="<?php se($user, 'id'); ?>" />
                                </td>
                                <td><?php se($user, "articles", "No Articles"); ?></td>
                            </tr>
                        <?php endforeach; ?>
                    </table>
                </td>
                <td>
                    <?php $counter = 0; ?>
                    <?php if (empty($active_articles)) : ?>
                        <p class="text-center mt-3">No Active Articles Found.</p>
                    <?php endif; ?>
                    <?php foreach ($active_articles as $article) : ?>
                        <?php $counter++; ?>
                        <div class="<?php if ($counter % 2 == 0) { se("bg-secondary bg-opacity-25"); } else { se("bg-secondary bg-opacity-50"); } ?>">
                            <label for="article_<?php se($article, 'id'); ?>"><?php se($article, 'title'); ?></label>
                            <input id="article_<?php se($article, 'id'); ?>" type="checkbox" name="articles[]" value="<?php se($article, 'id'); ?>" />
                        </div>
                    <?php endforeach; ?>
                </td>
            </tr>
        </tbody>
    </table>
</form>

```

Shows submitting button code on top (code for submitting the checkboxes...) the checkboxes have a value assigned of id's which are submitted upon click as a POST method. UCID/Date on top.

Checklist Items (1)

#5 Code related to submitting the checkbox lists

Task #3 - Points: 1

Text: Explain the solution

^COLLAPSE ^

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Describe the steps for the search and how it works for users and entities
<input checked="" type="checkbox"/> #2	1	Mention how you built the UI for this
<input checked="" type="checkbox"/> #3	1	Describe the steps for the associate/unassociate logic for the combination of users and entities

Response:

For this page the query for getting all the articles was pretty simple as it just got all the article titles and id's where the article was set to being active in the `is_active` column. There's also just one filter on the title that uses a % wildcard looking for the user's entered text in the title column of the articles (it's also limited to 25 when a search is done and also when no filter is done).

For the users query it was the same as the roles page, but just adjusted it to select the `UserArticles` table instead. In the query firstly it selects all the usernames and their id's from the `Users` table where the `username` contains what the user typed in which is done utilizing the % wildcard. The number of users are also limited to 25. However, this query becomes more complex as it has a sub select using `GROUP_CONCAT` to get the articles the user is associated with from the `UserArticles` table and the `is_active` is true/1. This sub select is done the same way as in other pages in which the `UserArticle` table is joined with the articletable where the article id's match and the user id in the association is of the user. This sub select is done for each user and puts it in each row of each user. The limit on the articles is also set to 25. Another important thing to note is that in this the title filter is also used to not just dump all the articles the user is associated with. If there's a filter on the title it will only show the id's of the ones that partially match the article title.

For the display there's a checkbox for each article and user. The checkbox holds the value of the id of the user or article that its labeled to. Upon submitting the form originally each article and user was looped though just negating the `is_active` column. However, this wasn't good as the user might go crazy and select 100+ articles and users resulting in that many queries. Therefore, it was changed and each user and article id associated is looped though and added to the params. Each value row has it's own user/article basically so everything can be done in just one query call. The unfavorite and favorite association works the same as before, it tries to insert the relationship, however if it exists the `is_active` column is negated.

Originally, this page was going to have an image and short description for each article, however it was not possible to create a clean layout (for me) so I just put the article titles and id's. To indicate what the random numbers are I also wrote it on the top of the columns "Article ID." Otherwise, the rest is self explanatory, it's just a table with two columns: user and titles. I also made the article titles alternate color to make it easy to differentiate.

Task #4 - Points: 1

Text: Add related links

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Include the heroku prod link for the page that creates the association
<input checked="" type="checkbox"/> #2	1	Add the pull request link for the branch related to this feature Note: the link should end with <code>/pull/#</code> . Same pull request shouldn't be used for each feature

URI #1

<https://github.com/Jhr-4/jhr4-IT202-008/pull/65>

URL #2

<https://github.com/Jhr-4/jhr4-IT202-008/pull/67>

URL #3

<https://jhr4-it202-008-prod-af7c79552123.herokuapp.com/Project/admin/assignFavorites.php>

Misc (1 pt.)

[^COLLAPSE ^](#)

Task #1 - Points: 1

Text: Screenshot of your project board from GitHub (tasks should be in the proper column)

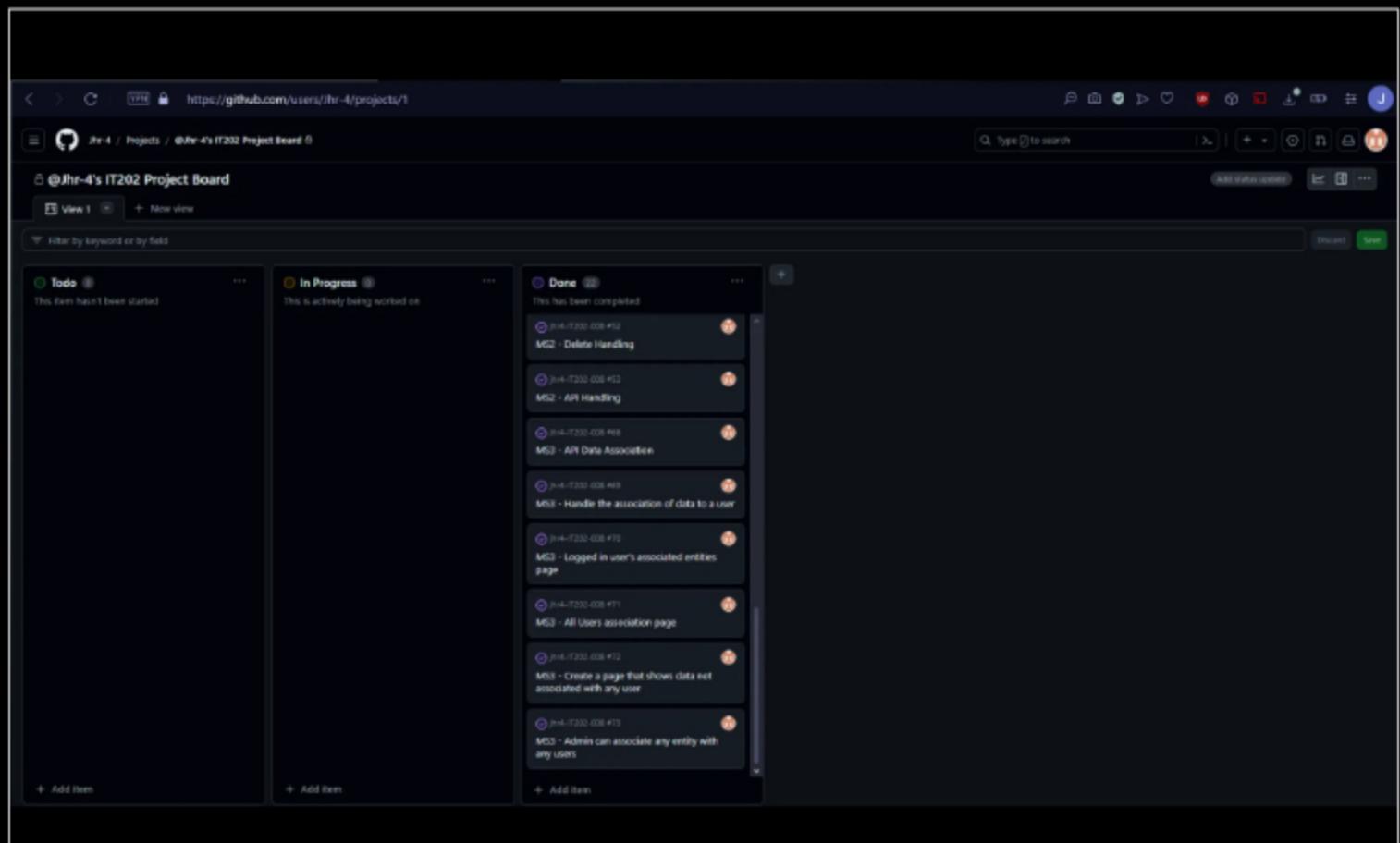
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



Project Board from GitHub, All tasks in Done.

Task #2 - Points: 1

Text: Provide a direct link to the project board on GitHub

MDL #1

Task #3 - Points: 1**Text:** Talk about any issues or learnings during this assignment**Response:**

I learned a lot in this assignment especially while working with sql and db's. While I already knew how to work with it from previous milestones I learned to filter my db calls and only select the data I want. For some of the parts my method was just get all the data and use php to sort, filter, and unset unnecessary items before displaying, however this wouldn't work as I had to show the total count. Therefore, it made me learn how to properly use sql to find what I'm looking for.

I had no issues really, it was just challenging to figure out how to solve the problems (code wise) which is expected.

Task #4 - Points: 1**Text:** WakaTime Screenshot**Details:****Task Screenshots:****Gallery Style: Large View****Small****Medium****Large**

WakaTime screenshot.

End of Assignment